

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ**

«На правах рукопису»
УДК 004.932.2

«До захисту допущено»
Завідувач кафедри СПСКС

(підпис) В.П.Тарасенко
(ініціали, прізвище)
“ ” _____ 2018р.

**Магістерська дисертація
на здобуття ступеня магістра**

зі спеціальності 123 Комп'ютерна інженерія
Комп'ютерні системи та компоненти

на тему СПОСОБИ РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ АВТОМОБІЛІВ У
СИСТЕМІ КОНТРОЛЬНО-ПРОПУСКНОГО ПУНКТУ

Виконав: студент II курсу, групи КВ-71мп
(шифр групи)

Кампов Владислав Олегович
(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник д.т.н., проф. Зайцев В.Г.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики
Кафедра системного програмування і спеціалізованих
комп'ютерних систем

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (спеціалізація) – 123 Комп'ютерна інженерія

Комп'ютерні системи та компоненти

ЗАТВЕРДЖУЮ

Завідувач кафедри

СПСКС

_____ В. П.

Тарасенко

«__» _____ 201

8 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Кампову Владиславу Олеговичу

1. Тема дисертації «Способи розпізнавання номерних знаків автомобілів у системі контрольно-пропускного пункту», науковий керівник дисертації Зайцев Володимир Григорович, д.т.н., проф, затверджені наказом по університету від «30» жовтня 2018 р. № 4030-с
2. Термін подання студентом дисертації «__» грудня 2018 р.
3. Об'єкт дослідження: є методи розпізнавання автомобільних номерів у системах КПП.
4. Предмет дослідження: є програмна реалізація системи розпізнавання автомобільних номерів у системі КПП.
5. Перелік завдань, які потрібно розробити:
 - провести аналіз методів розпізнавання тексту на зображеннях;
 - провести аналіз методів обробки зображень;
 - розробити метод розпізнавання автомобільних номерів на зображенні;
 - розробити програмне забезпечення для демонстрації роботи методу;
 - проаналізувати результати роботи моделі.
6. Орієнтовний перелік публікацій:
 - Тези доповіді «Прикладна математика та комп'ютинг» ПМК-2018-2

- Тези доповіді на V Міжнародна науково-технічна Internet-конференція

7. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання «04» жовтня 2017 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Грунтовне ознайомлення з предметною галуззю	15.10.2017	
2.	Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури, патентний пошук	07.03.2018	
3.	Робота над першим розділом магістерської дисертації; проведення наукового дослідження	15.04.2018	
4.	Проведення наукового дослідження; робота над другим розділом магістерської дисертації; розроблення програмного забезпечення	11.05.2018	
5.	Проведення наукового дослідження; робота над статтею за результатами наукового дослідження	08.06.2018	
6.	Проведення наукового дослідження; робота над третім розділом магістерської дисертації	13.07.2018	
7.	Завершення роботи над основною частиною магістерської дисертації; підготовка матеріалів доповіді на конференції ПМК-2018	17.10.2018	
8.	Оформлення текстової і графічної частини магістерської дисертації	09.11.2018	

Студент

В. О. Кампов

Науковий керівник дисертації

В. Г. Зайцев

РЕФЕРАТ

Актуальність теми. Системи розпізнавання тексту вже давно оточують нас у повсякденному житті. Однією з таких є системи розпізнавання автомобільних номерів. Такі системи потрібні в самих різних галузях. Наприклад, вони застосовуються у роботі автотранспортних підприємств, станцій техобслуговування, автомобільних парковок, тощо. Подібні системи дозволяють контролювати наявність автомобілів в зоні обслуговування, визначати час обслуговування автомобілів клієнтів, кількість вільних місць на парковці, фіксувати час перебування автомобіля в конкретній зоні, організовувати автоматичний в'їзд і виїзд автомобілів і т.д. Крім того, можливість автоматичного розпізнавання номера автомобіля є важливим аспектом контролю і забезпечення безпеки дорожнього руху з огляду на постійно зростаючу кількість автомобілів на дорогах.

Об'єктом дослідження є методи та алгоритми системи розпізнавання тексту.

Предметом дослідження є програмна реалізація розробленого алгоритму розпізнавання тексту.

Мета роботи: розробка й оптимізація алгоритму розпізнавання тексту задля підвищення її ефективності.

Методи дослідження. В роботі використовуються методи оптимізації, методи системного аналізу.

Наукова новизна полягає в наступному:

Запропоновано новий, оптимізований алгоритм розпізнавання тексту у системах контрольно-пропускного пункту, який відрізняється від існуючих підвищеною точністю розпізнавання тексту.

Практична цінність отриманих в роботі результатів полягає в тому, що запропонований алгоритм розпізнавання тексту у системах контрольно-пропускного пункту підвищує ключові показники ефективності розпізнавання тексту, а саме точність розпізнавання.

Апробація роботи. Основні положення і результати роботи були представлені та обговорювались на XI науковій конференції молодих вчених «Прикладна математика та комп'ютинг» ПМК-2018-2 (Київ, 14-16 листопада 2018 р.). та на V Міжнародній науково-технічній Internet-конференції (Київ, 22-23 листопада 2018 р.)

Структура та обсяг роботи. *Магістерська дисертація складається з вступу, чотирьох розділів, висновків та додатків. (може змінюватись)*

У вступі надано загальну характеристику роботи, виконано оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень, сформульовано мету дослідження, показано наукову новизну отриманих результатів і практичну цінність роботи.

У першому розділі розглянуто способи розпізнавання тексту у системах контрольно-пропускного пункту і основні його методи; наведено теоретичні засади розпізнавання, а також проведено порівняльний аналіз методів розпізнавання тексту у системах контрольно-пропускного пункту.

У другому розділі проаналізовано критерії оптимізації базового алгоритму; досліджено методи оптимізації базового алгоритму; розглянуто варіанти реалізації розробки базового алгоритму розпізнавання тексту.

У третьому розділі розроблено та описано етапи алгоритму розпізнавання тексту у системах контрольно-пропускного пункту; розроблено та описано структуру алгоритму розпізнавання тексту у системах контрольно-пропускного пункту; описано готовий розроблений розпізнавання тексту у системах контрольно-пропускного пункту.

У четвертому розділі з безпекою у приміщенні, в якому проводилась розробка, програмно реалізовано розроблений алгоритм розпізнавання тексту у системах контрольно-пропускного пункту; протестовано розроблений алгоритм розпізнавання тексту у системах контрольно-пропускного пункту; проаналізовано результати тестування розробленого алгоритму розпізнавання тексту у системах контрольно-пропускного пункту.

У висновках проаналізовано отримані результати роботи.

Робота виконана на 80 аркушах, містить 2 додатки та посилання на список використаних літературних джерел з 15 найменувань. У роботі наведено 41 рисуноків та 3 таблиці.

Ключові слова: розпізнавання тексту, автомобільні номери, контрольно-пропускний пункт

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ **Ошибка! Закладка не определена.**

ВСТУП	8
1. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ РОЗПІЗНАВАННЯ ТЕКСТУ У ЗОБРАЖЕННЯХ	10
1.1 Основні поняття теорії розпізнавання образів	10
1.2 Методи розпізнавання тексту	19
2. АНАЛІЗ МЕТОДІВ РОЗПІЗНАВАННЯ ТЕКСТУ	25
2.1 Опис інструментарію	25
2.2 Методи розпізнавання тексту на зображеннях	28
2.4 Використання рекурентних нейронних мереж	53
3. АЛГОРИТМ РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРНИХ ЗНАКІВ У СИСТЕМІ КОНТРОЛЬНО ПРОПУСКНОГО ПУНКТУ	56
3.1 Покращення візуальної якості зображень шляхом поелементного перетворення	56
3.2 Локалізація, Нормалізація, Сегментація	69
3.2 Навчання та використання рекурентної нейронної мережі	77
4. ТЕСТУВАННЯ АЛГОРИТМУ РОЗПІЗНАВАННЯ ТЕКСТУ У СИСТЕМІ КОНТРОЛЬНО-ПРОПУСКНОГО ПУНКТУ	81
4.1 Особливості реалізації системи	81
4.2 Тестування системи	84

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

ДСТУ – державний стандарт України

КМ – комп'ютерна мережа

КС – комп'ютерна система

НМ – нейрона мережа

ОС – операційна система

ПЗ – програмне забезпечення

ПСІ – прикладний програмний інтерфейс

СШН – схований шар нейронів

CLI – інтерфейс командної строки (Command Line Interface)

DCT – Дискретне косінусне перетворення

ВСТУП

На даний момент наше життя важко уявити без вже здобутого рівня комп'ютеризації навколишнього середовища. З розвитком технологій людство спромоглося автоматизувати чимало процесів, які ще зовсім нещодавно потребували безпосереднього втручання людини. Інформаційні технології переслідують нас усюди – від купівлі проїзного квитка у транспорті до використання окулярів доповненої реальності у важкій промисловості. Однією з таких технологій є і так званий комп'ютерний зір.

Комп'ютерне бачення являє собою теорію та технологію створення комп'ютерних систем, що отримують будь-яку інформацію у вигляді зображень. Це можуть бути зображення з відеокамер, відеореєстраторів або навіть трьохвимірне зображення з медичного сканера. Одним з підрозділів комп'ютерного зору є і розпізнавання образів – предметів, тексту, тощо. Наприклад, розпізнавання автомобільних номерних знаків є досить вузькою, але дуже важливою галуззю у наш час.

В даний час системи автоматичного розпізнавання автомобільних номерів потрібні в самих різних галузях. Подібні системи дозволяють контролювати наявність автомобілів в зоні обслуговування, визначати час обслуговування автомобілів клієнтів, кількість вільних місць на парковці, фіксувати час перебування автомобіля в конкретній зоні, організовувати автоматичний в'їзд і виїзд автомобілів і т.д. Крім того, можливість автоматичного розпізнавання номера автомобіля є важливим аспектом контролю і забезпечення безпеки дорожнього руху з огляду на постійно зростаючу кількість автомобілів на дорогах.

В магістерській дисертації розглянуто алгоритм, створений на базі існуючих з використанням модифікацій, що дозволить більш ефективно розпізнавати автомобільні номерні знаки і призведе до підвищення якості

програмного продукту, а отже його подальшого розвитку і розповсюдження.

1. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ РОЗПІЗНАВАННЯ ТЕКСТУ У ЗОБРАЖЕННЯХ

1.1 Основні поняття теорії розпізнавання образів

Розпізнавання образів є однією з фундаментальних проблем теорії інтелектуальних систем. З іншого боку, задача розпізнавання образів має величезне практичне значення. Замість терміна “розпізнавання” часто вживається інший – “класифікація”. Ці два терміни у багатьох випадках розглядаються як синоніми, але не є повністю взаємозамінюваними. Кожен з них має свої сфери застосування, і інтерпретація обох термінів часто залежить від специфіки конкретної задачі. Теорія розпізнавання образів — розділ кібернетики, що розвиває теоретичні основи й методи класифікації і ідентифікації предметів, явищ, процесів, сигналів, ситуацій і т. п. об'єктів, які характеризуються скінченним набором деяких властивостей і ознак. Такі задачі вирішуються досить часто, наприклад, при переході або проїзді вулиці за сигналами світлофора.

Наведемо деякі типові постановки задач розпізнавання.

1. Задача ідентифікації, яка полягає в тому, щоб вирізнити певний конкретний об'єкт серед йому подібних (наприклад, впізнати серед інших людей свою дружину).

Віднесення об'єкта до того чи іншого класу або класифікація. Це може бути, наприклад, задача розпізнавання літер або прийняття рішення про наявність дефекту у деякій технічній деталі. Віднесення об'єкта до певного класу відображає найтипівішу проблему класифікації, і, коли говорять про розпізнавання образів, найчастіше мають на увазі саме цю проблему. Тому ми розглядатимемо її першочергово.

Кластерний аналіз, який полягає в поділі заданого набору об'єктів на класи — групи об'єктів, схожі між собою за тим чи іншим критерієм. Цю задачу часто називають класифікацією без учителя, оскільки, на відміну від задачі 2, класи апіорно не задані

Розпізнавання образів (об'єктів, сигналів, ситуацій, явищ або процесів) – завдання ідентифікації об'єкта або визначення будь-яких його властивостей за його зображенням (оптичне розпізнавання) або аудіозапису (акустичне розпізнавання) та іншими характеристиками.

Одним з базових понять є поняття множини. У комп'ютері множина представляється набором неповторюваних однотипних елементів. Слово "неповторюваних" означає, що якийсь елемент у множині або є, або його там немає. Універсальна множина включає всі можливі для розв'язуваної задачі елементи, порожня не містить жодного.

Методика віднесення елемента до якогось образу називається вирішальним правилом. Ще одне важливе поняття – метрика, спосіб визначення відстані між елементами універсальної множини. Чим менша ця відстань, тим більш схожими є об'єкти (символи, звуки та ін.) – те, що ми розпізнаємо. Звичайно елементи задаються у вигляді набору чисел, а метрика – у вигляді функції. Від вибору представлення образів і реалізації метрики залежить ефективність програми. Один і той самий алгоритм розпізнавання з різними метриками буде помилятися з різною частотою.

Навчанням зазвичай називають процес вироблення в деякій системі тієї чи іншої реакції на групи зовнішніх ідентичних сигналів шляхом багаторазового впливу на систему зовнішньої корегування. Таке зовнішнє коригування у навчанні прийнято називати "заохоченнями" і "покараннями".

Механізм генерації цього корегування практично повністю визначає алгоритм навчання. Самонавчання відрізняється від навчання тим, що тут додаткова інформація про вірність реакції системі не повідомляється.

Розпізнавання кольору лампи світлофора, що засвітилася, і знання правил дорожнього руху дозволяє прийняти правильне рішення про те, можна, чи не можна переходити вулицю в цей момент. У більшості випадків, сприймаючи явища навколишнього світу, людина здійснює їх

класифікацію, тобто розбиває ці явища (предмети, ситуації) на групи схожих явищ (саме схожих, а не тотожних). По тим чи іншим причинам необхідно віднести в одну групу у чомусь "подібні" явища чи предмети, які можуть при цьому значно відрізнятися один від одного. Наприклад, усі фігури, зображені на наступному рис. 1.1, ми називаємо "літерою А", незважаючи на їх значну відмінність у їх написанні.



Рисунок 1.1 – Відмінність у написанні літери А

Суттєвим є той факт, що виділивши такі групи (множини) об'єктів, ми отримуємо здатність "упізнавати", тобто встановлювати належність до вже відомої множини, нові об'єкти, які ще не зустрічалися нам раніше, наприклад впізнавати букви, написані новим для нас почерком. Отримавши уявлення про те, що являє собою буква "А" на основі деякої, зазвичай невеликої кількості екземплярів цієї букви, ми спроможні упізнати як завгодно велику кількість інших її екземплярів.

Проте, далеко не всі множини об'єктів дають змогу на основі невеликої частини множини упізнати як завгодно багато інших невідомих нам її представників. Наприклад, фотографії студентів деякого вузу утворюють множину. Проте неможливо після ознайомлення зі скажімо десятьма фотографіями студентів визначити по новій фотографії є людина студентом цього вузу чи ні. Таким чином існують множини деякого особливого типу. Ці множини мають характерну властивість, яка виявляється у тому, що після ознайомлення із скінченною частиною об'єктів цих множин, можна упізнавати як завгодно велику кількість інших їх представників. Множини такого типу будемо називати образами.

Прикладами образів можуть бути такі множини: чоловіки, дитячі портрети, ссавці, картини Пікассо, цифри 5, зображення винищувачів МІГ.

Застосовувати до них термін "образ" ми можемо тому, що ознайомлення з образом не пов'язано із запам'ятовуванням окремих об'єктів, а упізнавання нового об'єкта відбувається без безпосереднього порівняння із кожним раніше відомим.

Характерна властивість образів об'єктивна у тому сенсі, що різні люди (живі істоти), які навчалися на різних групах об'єктів образу, у переважній більшості однаково і незалежно один від одного класифікують одні і ті самі нові об'єкти. Саме об'єктивність цієї властивості образів дає змогу людям, які вчилися у різних школах, успішно розпізнавати раніше невідомий їм почерк.

Проте, самі образи у той самий час є у певній мірі "розпливчастими" і питання про належність об'єкта до даного образу не завжди може бути вирішене однозначно. На Рис. 2 показано, як за допомогою невеликих послідовних змін можна перетворити цифру 5 у цифру 3. рис. 1.2 ілюструє важливу особливість об'єктів кожного образу. Можна в деяких межах змінювати об'єкт, але він усе одно залишиться об'єктом того самого образу.

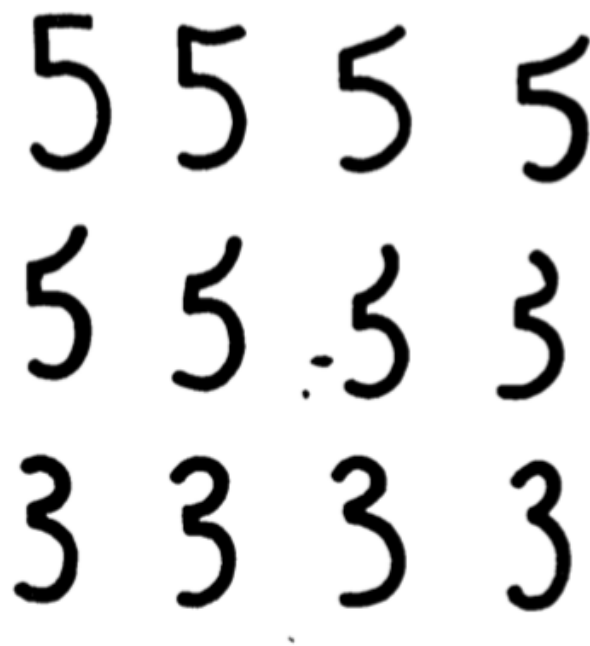


Рисунок 1.2 – Перетворення цифри 5 у цифру 3

Створення штучних систем розпізнавання образів залишається складною теоретичною й технічною проблемою. Необхідність у такому розпізнаванні виникає в самих різних областях — від військової справи й систем безпеки до оцифрування різних аналогових сигналів. Сприйняття явищ у формі образів відіграє надзвичайно важливу роль у процесах пізнання зовнішнього світу. У процесі біологічної еволюції багато тварин за допомогою зорового й слухового апарата вирішили задачу розпізнавання образів досить добре. Як впливає із самого означення образу "розпізнаванню" нових для нас об'єктів передуює процес навчання. Під час навчання істоти ознайомлюються із деякою кількістю об'єктів і, крім цього, із якогось джерела (наприклад від батьків, старших і т. п.) отримують інформацію про те, до якого образу відноситься кожний із цих об'єктів. Цей процес отримав назву "навчання з учителем". Більш загальний характер має "навчання без учителя", у процесі якого система вчиться спонтанно виконувати поставлене завдання без втручання з боку "вчителя". Навчання машин без вчителя формулюється як задача кластерного аналізу. Вибірка об'єктів розбивається на кластери (множини, що мають порожній перетин), таким чином, що кожний кластер складається із "схожих" об'єктів, а різні кластери "суттєво" відрізняються один від одного. Кластеризація часто використовується в якості допоміжного засобу розв'язування задач класифікації та регресійного аналізу. Деякі алгоритми розв'язування задач класифікації комбінують навчання з учителем та навчання без учителя (наприклад навчання мереж векторного квантування).

Традиційно задачі розпізнавання образів включають у коло задач штучного інтелекту. Можна виділити два основних напрямки:

- Вивчення здібностей до розпізнавання, якими володіють живі істоти, їхнє пояснення й моделювання;

Розвиток теорії й методів побудови пристроїв, призначених для

розв'язання окремих задач у прикладних цілях.

Розпізнавання образів — це віднесення вихідних даних до певного класу за допомогою виділення істотних ознак, що характеризують ці дані, із загальної маси несуттєвих даних. При постановці задач розпізнавання намагаються користуватися математичною мовою.

Для оптичного розпізнавання образів можна застосувати метод перебору вигляду об'єкта під різними кутами, масштабами, зсувами й т. д. Для букв потрібно перебирати шрифт, властивості шрифту й т. д. Другий підхід — знайти контур об'єкта й досліджувати його властивості (зв'язність, наявність кутів і т. д.)

Ще один підхід — використовувати штучні нейронні мережі (багатошарові перцептрони, мережі квантування, мапи Кохонена, рекурентні мережі). Цей метод вимагає або великої кількості прикладів задачі розпізнавання (із правильними відповідями), або спеціальної структури нейронної мережі, що враховує специфіку даної задачі.

Ф. Розенблатт уводячи поняття про модель мозку, завдання якої полягає в тому, щоб показати, як у деякій фізичній системі, структура й функціональні властивості якої відомі, можуть виникати психологічні явища та описав найпростіші експерименти з розрізнення. Дані експерименти цілком стосуються до методів розпізнавання образів, але відрізняються тим, що алгоритм розв'язання не детермінований. Найпростіший експеримент, на основі якого можна одержати психологічно значиму інформацію про деяку систему, зводиться до того, що моделі пред'являються два різних стимули й потрібно, щоб вона реагувала на них різним чином. Метою такого експерименту може бути дослідження можливості спонтанного розрізнення стимулів системою при відсутності втручання з боку експериментатора, або, з навпаки, вивчення примусового розрізнення, при якому експериментатор прагне навчити систему здійснювати необхідну класифікацію. У досвіді з навчанням перцептрону зазвичай пред'являється деяка послідовність образів, у яку

входять представники кожного із класів, що підлягають розрізненню. Відповідно до деякого правила модифікації пам'яті правильний вибір реакції підкріплюється. Потім персептрону пред'являється контрольний стимул і визначається ймовірність одержання правильної реакції для стимулів даного класу. Залежно від того, збігається чи не збігається обраний контрольний стимул з одним з образів, які використовувалися в навчальній послідовності, отримують різні результати:

Якщо контрольний стимул не збігається з жодним із навчальних стимулів, то експеримент пов'язаний не тільки з чистим розрізненням, але містить у собі й елементи узагальнення.

Якщо контрольний стимул збуджує деякий набір сенсорних елементів, цілком відмінних від тих елементів, які активізувалися при впливі раніше пред'явлених стимулів того ж класу, то експеримент є дослідженням чистого узагальнення.

Персептрони не мають здатності до чистого узагальнення, але вони цілком задовільно функціонують в експериментах із розрізнення, особливо якщо контрольний стимул досить близько збігається з одним з образів, щодо яких персептрон уже нагромадив певний досвід.

Приклади задач розпізнавання образів:

- Розпізнавання літер
- Розпізнавання штрих-кодів
- Розпізнавання автомобільних номерів
- Розпізнавання осіб
- Розпізнавання мови
- Розпізнавання зображень
- Розпізнавання локальних ділянок земної кори, у яких знаходяться родовища корисних копалин

Розпізнавання образів застосовується в наступних областях:

- Біоінформатика: пошук шаблонів в ДНК
- Бази даних: пошук і класифікація

Обробка текстів: тематична класифікація

Аналіз зображень: розпізнавання символів, робота з картами, розпізнавання осіб, поділ об'єктів

Виробництво: контроль якості (візуальна перевірка коректності мікросхем)

- Пошук по мультимедіа: визначення жанрів
- Біометрія: ідентифікація людини за відбитками пальців, по райдужній оболонці ока

Прогнозування: погода, сейсмологія, геологія

Обробка мови: переклад аудіо-сигналів в текст

Повернемося до означення. Розпізнавання образів — процес віднесення об'єкта з фіксованою групою його властивостей до одного об'єкту з множини образів за заздалегідь обумовленим правилом. Наприклад, рибу у тенетах треба поділити на окунів і лососів. Припустимо, що це робиться по довжині риби. Тобто у нас є об'єкт "риба", і за значенням властивості "довжина" ми відносимо рибу або до образу "лосось", або до образу "окунь".

Виділимо найбільш важливі кроки у процедурі розпізнавання:

Сприйняття образу. На цьому етапі проводять отримання значень характеристичних властивостей об'єкта (вимірювання лінійних вимірів, фотографування, оцифровка звуку).

Попередня обробка. Видалення шумів, представлення зображення в чорно-білому варіанті, обрізання непотрібних частин зображення.

1. Виділення характеристик (індексація) . На цьому етапі вимірюються характеристичні властивості об'єкта (вимірюємо довжину риби та її колір).
2. Класифікація (прийняття рішення).

Розробка системи розпізнавання:

1. Отримати навчальну вибірку (тренувальну колекцію) (навчальна

вибірка — множина об'єктів, для яких відомі їхні образи. Наприклад колекція аудіо записів для кожного звуку, або колекція зображень кожної букви латинського алфавіту).

2. Вибрати модель представлення об'єктів.
3. Вибрати значущі характеристики. Це один з найважливіших етапів розробки системи розпізнавання. Наприклад, якщо у випадку ідентифікації риби окунь / лосось в якості характеристики вибрати тільки довжину риби, то ніяке правило класифікації не зможе точно визначити тип риби, оскільки можна зустріти лосося і окуня однакової довжини.
4. Розробити правило класифікації. Правило класифікації — правило, яке за значеннями характеристикних властивостей об'єкта відносить його до одного з образів.

Навчання розпізнавальної системи. На цьому етапі алгоритм навчання "збирає досвід" на основі розпізнавання навчальної вибірки. Для того, щоб правильно виставити коефіцієнти (параметри) розпізнавальної системи, алгоритм навчання застосовують на навчальній вибірці, контролюючи результат роботи алгоритму.

Перевірка якості навчання. Повернення назад до кроків 2 (3, 4) ... Якщо частота помилок алгоритму не влаштовує, то необхідно повернутися до п. 2, 3, 4. Інтуїтивно зрозуміло, що збільшення кількості характеристикних властивостей, збільшення тренувальної колекції покращують якість роботи алгоритму.

Оптимізація розпізнавальної системи.

Після того, як якість роботи системи розпізнавання підходить під умову розглянутої задачі, іноді доводиться провести його оптимізацію. Початковий алгоритм розпізнавання може бути занадто довгим або ресурсоємним. Прискорити процедуру розпізнавання можна зменшивши кількість характеристикних властивостей об'єкта, вибравши інші

характеристичні властивості, використовуючи інше правило класифікації.

1.2 Методи розпізнавання тексту

Виділяють 4 групи методів розпізнавання:

1. Порівняння із зразком. Застосовуємо геометричну нормалізацію і вважаємо відстань до прототипу. Найбільш наочно застосування цього методу в розпізнаванні тексту. Завдання. У нас є зображення відсканованого символу і колекція зображень зразків (усіх букв абетки) , ми хочемо визначити, який букві алфавіту відповідає відскановане зображення. Розв'язок. Масштабуємо зображення символу до розмірів зразків і виберемо той, відстань до якого мінімальна.
2. Нейронні мережі. Вибираємо вид мережі і налаштовуємо коефіцієнти. На вхід нейронної мережі подається об'єкт для розпізнавання. Група рецепторів мережі відповідає за прийом своєї характеристичної властивості.
3. Статистичні методи.

Структурні та синтаксичні методи. Розбираємо об'єкт на елементи. Будуємо правило, в залежності від входження окремих елементів та їх послідовностей.

Наприклад, припустимо, нам необхідно визначити вид риби, поданої на вхід. Спростимо задачу до випадку 2 видів (окунь і лосось). Перший розв'язок, який приходить в голову — зробити припущення про вигляд риби по її довжині (рис. 1.3).

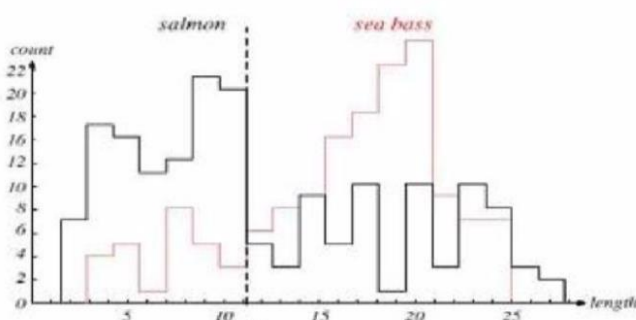


Рисунок 1.3 – Ілюстрація припущення про вигляд риби по її довжині

Інша характеристична властивість — забарвлення (рис. 1.4).

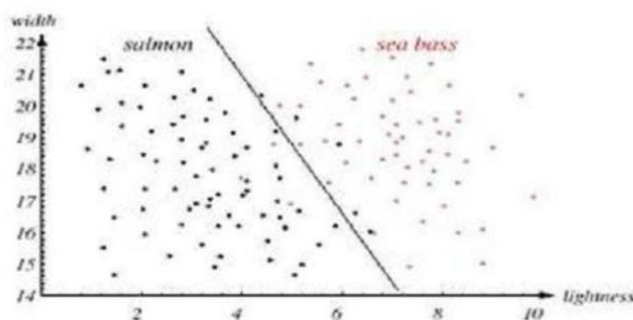


Рис 1.4 – Ілюстрація припущення про вигляд риби по її забарвленню

Керуючись лише однією характеристикою не можливо точно визначити вид риби. Але керуючись обома ознаками, можна визначати вид риби більш точно з використанням гіперплощини.

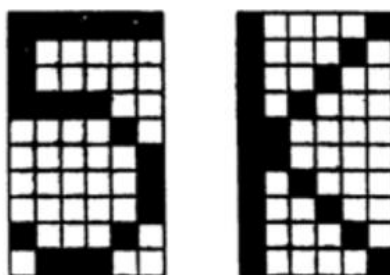


Рисунок 1.5 – Символи 5 і К

Розглянемо перетворення зображення у машинний код. Зображення розташовується у деякому прямокутному полі, розбитому на однакові квадрати (елементи поля або рецептори). Квадрати, на які потрапило зображення, цілком зафарбовуються у чорний колір і позначаються одиницею. Інші елементи позначаються нулем. Якщо проаналізувати елементи зліва направо і зверху вниз, то цифри 5 та літери "К" на рис. 1.5 будуть відповідати кодині на рис. 1.6

«5» — 11111110000010000011110000001000000100000100000
1100010011100
«К» — 1000011000101001001010001100001100001010001001
00100010100001.

Рисунок 1.6 – Переведення символів до машинного коду

відповідно до рис. 1.4

У загальному випадку код фігури записується у вигляді x_1, x_2, \dots, x_n , де $x_i \in \{0, 1\}$, $i = \overline{1, n}$, n — кількість елементів поля рецепторів.

Точка у просторі рецепторів називається граничною точкою множини, якщо у її коді є хоча б одна цифра, зміна якої переводить точку у іншу множину. У протилежному випадку точка називається внутрішньою. Якщо для множини точок виконуються умови

а) число граничних точок невелике порівняно із їх загальним числом,

б) довільні дві внутрішні точки множини можуть бути з'єднані достатньо плавною лінією, яка проходить лише через внутрішні точки,

в) майже кожна внутрішня точка має достатньо великий окіл, який цілком міститься у множині, то така множина точок називається компактною множиною.

Описаний метод кодування не враховує колір і тому придатний лише для чорно-білих зображень. У загальному випадку кожний розряд коду може приймати значення з деякого відрізка $[a, b]$.

Таким чином ми приходимо до гіпотези – простому зоровому образу відповідає компактна множина точок у просторі рецепторів. Поняття компактності проілюстровано на рис. 1.7.

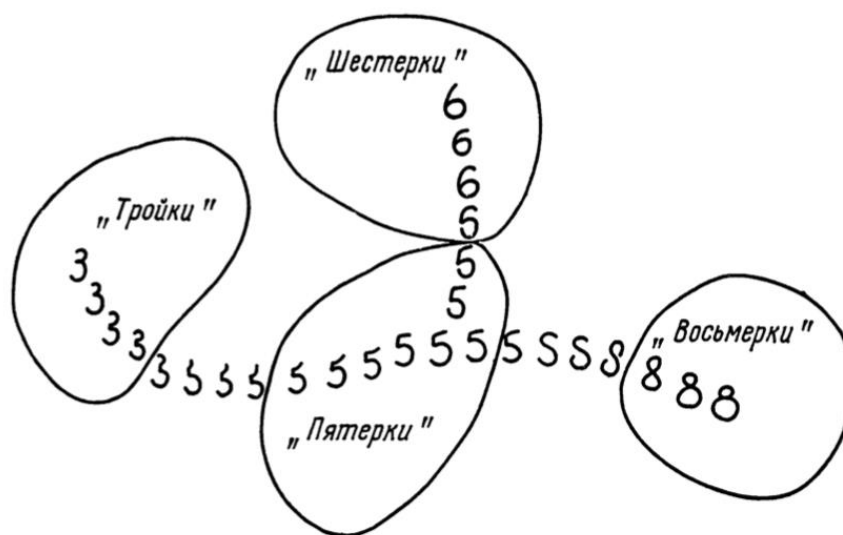


Рисунок 1.7 – Ілюстрація поняття компактності

Гіпотеза компактності дає змогу звести задачу навчання розпізнавання образів до задачі побудови відокремлюючих поверхонь по відомим точкам кожного образу. У результаті простір рецепторів розбивається на кілька областей, про кожну з яких відомо, що вона переважно містить точки, які відповідають певному образу. Якщо тепер потрібно розпізнати нову невідому фігуру, то потрібно перетворити цю фігуру у точку простору рецепторів і визначити, у якій частині простору ця точка розташована.

Розглянемо метод кластеризації. У багатьох прикладних задачах вимірювати ступінь подібності об'єктів істотно простіше, ніж формувати опис за ознаками. Наприклад, набагато легше порівняти дві фотографії і сказати, що вони належать одній людині, ніж зрозуміти, на підставі яких ознак вони схожі.

Завдання класифікації об'єктів на основі їх подібності між собою, коли належність навчальних об'єктів до будь-яких класів не задається, називається завданням кластеризації. В основному розглядаються статистичні, ієрархічні та графові алгоритми кластеризації. Існують методи багатовимірного шкалювання, що дозволяють відновлювати опис за ознаками об'єктів за матрицею попарних відстаней між ними.

Задача кластеризації (або навчання без вчителя) полягає в наступному. Є навчальна вибірка $Xl = \{x_1, \dots, x_l\} \subset X$ і функція відстані між об'єктами $\rho(x, x')$.

Потрібно розбити вибірку на непересічні підмножини, які називаються кластерами, так, щоб кожен кластер складався з об'єктів, близьких за метрикою ρ , а об'єкти різних кластерів істотно відрізнялися. При цьому кожному об'єкту $x_i \in Xl$ приписується мітка (номер) кластера u_i . Алгоритм кластеризації – це функція $a: X \rightarrow Y$, яка будь-якому об'єкту $x \in X$ ставить у відповідність мітку кластера $u \in Y$.

Множина міток Y в деяких випадках відома заздалегідь, однак частіше ставиться завдання визначити оптимальну кількість кластерів з точки зору того чи іншого критерію якості кластеризації. Рішення задачі кластеризації принципово неоднозначне, і цьому є кілька причин. По-перше, не існує однозначно найкращого критерію якості кластеризації. Відомий цілий ряд досить розумних критеріїв, а також ряд алгоритмів, які не мають чітко вираженого критерію, але здійснюють досить розумну кластеризацію «з побудови». Всі вони можуть давати різні результати. По-друге, число кластерів, як правило, невідоме заздалегідь і встановлюється відповідно до деякого суб'єктивного критерію. По-третє, результат кластеризації істотно залежить від метрики ρ , вибір якої, як правило, також суб'єктивний і визначається експертом.

Кластеризація (навчання без вчителя) відрізняється від класифікації (навчання з учителем) тим, що мітки вихідних об'єктів u_i спочатку не задані, і навіть може бути невідома сама множина Y . У цьому сенсі задача кластеризації ще більшою мірою некоректно поставлена, ніж задача класифікації. Цілі кластеризації можуть бути різними в залежності від особливостей конкретної прикладної задачі:

- Зрозуміти структуру множини об'єктів Xl , розділивши її на групи схожих об'єктів. Спростити подальшу обробку даних і прийняття рішень, працюючи з кожним кластером окремо

(стратегія «розділяй і володарюй»).

- Скоротити обсяг збережених даних в разі надвеликої вибірки X_1 , залишивши по одному найбільш типовому представникові від кожного кластеру.
- Виділити нетипові об'єкти, які не підходять до жодного з кластерів.

Цю задачу називають однокласовою класифікацією, виявленням нетиповості або новизни (novelty detection). У першому випадку кількість кластерів намагаються зробити меншою. У другому випадку важливіше забезпечити високу ступінь подібності об'єктів усередині кожного кластера, а кластерів може бути скільки завгодно. У третьому випадку найбільший інтерес представляють окремі об'єкти, які не вписуються в жоден з кластерів. У всіх цих випадках може застосовуватися ієрархічна кластеризація, коли великі кластери дробляться на більш дрібні, ті в свою чергу дробляться на ще дрібніші, і т. ін. Такі завдання називаються завданнями таксономії (taxonomy). Результатом таксономії є не просте розбиття множини об'єктів на кластери, а деревовидна ієрархічна структура. Замість номера кластера об'єкт характеризується перерахуванням всіх кластерів, яким він належить, від великого до дрібного. Класичним прикладом таксономії на основі подібності є систематизація живих істот, запропонована Карлом Ліннеєм в середині XVIII століття. У сучасному уявленні біологічна ієрархія має близько 30 рівнів, 7 з них вважаються основними: царство, тип, клас, відділ, сімейство, рід, вид. Таксономії будуються у багатьох областях знань, щоб упорядкувати інформацію про велику кількість об'єктів. Саме алгоритми ієрархічної кластеризації дозволяють автоматизувати процес побудови таксономій.

Типи кластерних структур. Формується реєстр різних типів кластерних структур, які можуть виникати в практичних задачах. Різні алгоритми кластеризації можуть бути більш-менш успішні в цих

ситуаціях. Прості алгоритми, як правило, вузько спеціалізовані і дають адекватні результати тільки в одній-двох ситуаціях. Більш складні алгоритми, такі як FOREL або агломеративна процедура Ланса-Вільямса, справляються з кількома типами ситуацій. Однак створення алгоритму, що успішно працює в усіх ситуаціях без винятку, представляється важкою і навряд чи можливою для розв'язання задачею.

2. АНАЛІЗ МЕТОДІВ РОЗПІЗНАВАННЯ ТЕКСТУ

2.1 Опис інструментарію

Операційною системою для розробки було обрано Ubuntu GNU/Linux.

Linux - сімейство Unix-подібних операційних систем на базі ядра Linux, що включають той чи інший набір утиліт і програм проекту GNU, і, можливо, інші компоненти. Як і ядро Linux, системи на його основі як правило створюються і поширюються відповідно до моделі розробки вільного та відкритого програмного забезпечення. Linux-системи поширюються в основному безкоштовно у вигляді різних дистрибутивів - у формі, готової для установки і зручною для супроводу і оновлень, - і мають свій набір системних і прикладних компонентів, як вільних, так можливо і власницьких.

Ubuntu – один з таких дистрибутивів. Ubuntu орієнтована на зручність і простоту використання. Вона включає широко поширене використання утиліти `udo`, яка дозволяє користувачам виконувати адміністраторські завдання, що не запускаючи потенційно небезпечну сесію суперкористувача.

Ubuntu, крім того, має розвинену інтернаціоналізацію, що забезпечує максимальну доступність для представників різних мовних груп.

Ubuntu для роботи рекомендується від 512 мегабайт RAM і, при установці на жорсткий диск, від п'яти гігабайт вільного простору, а

гранично мінімальні вимоги набагато нижче. Як вже було зазначено, для розробки системи використано мову програмування Python.

Python – високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python – високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує кілька парадигм програмування, в тому числі структурне, об'єктно-орієнтоване, функціональне, імперативне і аспектно-орієнтоване програмування. Основні архітектурні риси - динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень і зручні високорівневі структури даних. Код в Python організовується у функції та класи, які можуть об'єднуватися в модулі (вони в свою чергу можуть бути об'єднані в пакети).

Еталонної реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ. Він поширюється під вільною ліцензією Python software Foundation License, що дозволяє використовувати його без обмежень в будь-яких додатках, включаючи пропрієтарні. Є реалізації інтерпретаторів для JVM (з можливістю компіляції), .NET (з можливістю компіляції), LLVM та інших. Проект PyPy пропонує реалізацію Python з використанням JIT-компіляції, яка значно збільшує швидкість виконання Python-програм.

Python – мова програмування, що активно розвивається, нові версії (з додаванням / зміною мовних властивостей) виходять приблизно раз в два з половиною роки. Внаслідок цього і деяких інших причин на Python

відсутні стандарт ANI, IO або інші офіційні стандарти, їх роль виконує CPython.

Оскільки Python — інтерпретована мова, математичні алгоритми, часто працюють в ньому набагато повільніше ніж у компільованих мовах, таких як C або навіть Java. NumPy намагається вирішити цю проблему для великої кількості обчислювальних алгоритмів забезпечуючи підтримку багатовимірних масивів і безліч функцій і операторів для роботи з ними. Таким чином будь-який алгоритм який може бути виражений в основному як послідовність операцій над масивами і матрицями працює також швидко як еквівалентний код написаний на C.

NumPy можна розглядати як гарну вільну альтернативу MATLAB, оскільки мова програмування MATLAB зовні нагадує NumPy: обидві вони інтерпретовані, і обидві дозволяють користувачам писати швидкі програми поки більшість операцій проводяться над масивами або матрицями, а не над скалярами. Перевага MATLAB у великій кількості доступних додаткових тулбоксів, включаючи такі як пакет `imulink`. Основні пакети, що доповнюють NumPy, це: `scipy` — бібліотека, що додає більше MATLAB-подібної функціональності; `Matplotlib` — пакет для створення графіки в стилі MATLAB. Внутрішньо як MATLAB, так і NumPy базується на бібліотеці LAPACK, призначеної для вирішення основних задач лінійної алгебри.

TensorFlow — відкрита програмна бібліотека для машинного навчання цілій низці задач, розроблена компанією Google для задоволення її потреб у системах, здатних будувати та тренувати нейронні мережі для виявлення та розшифровування образів та кореляцій, аналогічно до навчання й розуміння, які застосовують люди. Її наразі застосовують як для досліджень, так і для розробки продуктів Google, часто замінюючи на його ролі її закритого попередника, DittBelief. TensorFlow було початково розроблено командою Google Brain для внутрішнього використання в

Google, поки її не було випущено під відкритою ліцензією Apache 2.0 9 листопада 2015 року.

TenorFlow забезпечує ППІ для Python, а також для C++, Haskell, Java та Go.

Серед застосувань, для яких TenorFlow є основою, є програмне забезпечення автоматизованого опису зображень, таке як DeepDream. 26 жовтня 2015 року Google офіційно реалізувала RankBrain, який підтримує TenorFlow. RankBrain тепер обробляє суттєве число пошукових записів, замінюючи та доповнюючи традиційні статичні алгоритми на основі результатів пошуку.

2.2 Методи розпізнавання тексту на зображеннях

Одне з найвдаліших формулювань ключової парадигми теорії розпізнавання таке: будь-який об'єкт у природі є унікальним, всі об'єкти є типізованими.

Зміст цієї парадигми такий. Кожний об'єкт характеризується тими чи іншими властивостями. Наявність чи відсутність таких властивостей, а також якісні та кількісні характеристики цих властивостей розглядаються як ознаки об'єкта. Унікальність будь-якого об'єкта означає те, що в природі не існує двох різних об'єктів, для яких збігаються абсолютно всі ознаки, а це дозволяє, принаймні теоретично, відрізнити один об'єкт від іншого. Але деякі ознаки різних об'єктів можуть збігатися, і це дає підстави говорити про те, що ці об'єкти належать до одного типу або класу.

Фундаментальні поняття “клас” та “об'єкт” неможливо повністю формалізувати. Спробуємо навести їх неформальні визначення. Об'єктом у теорії розпізнавання прийнято називати будь-яку сутність, що існує або могла б існувати в реальному світі, а також будь-яке явище або процес.

Це дуже широке визначення, подальші уточнення можуть бути пов'язані з тим чи іншим звуженням нашого розуміння про те, що саме

слід вважати об'єктом. Так, реально існуюча Ейфелева вежа буде вважатися об'єктом практично у будь-якій інтерпретації, а “розвиток наукових досліджень про етнокультурний стан Ефіопії та Антарктиди за період з липня 1898 року по січень 1927 року” – можливо, ні. Тут усе залежить від специфіки конкретної задачі і від мети, з якою вона вирішується

Класом у теорії розпізнавання образів прийнято називати сукупність об'єктів, які мають ті чи інші спільні ознаки.

Клас може об'єднувати фізично існуючі сутності (наприклад, людина, яблуко) або бути абстрактним поняттям (горе, економічний крах і т.п.).

Ознаки, що дають можливість відрізнити представників одного класу від іншого, прийнято називати інформативними ознаками.

Ознаки, спільні для всіх представників класу, називатимемо інваріантами класу.

Іноді вживають формальніше визначення класу: класом називається сукупність об'єктів, пов'язаних між собою деяким відношенням еквівалентності, або, в крайньому разі, толерантності. Нагадаємо, що відношенням еквівалентності називається відношення, яке є симетричним, рефлексивним і транзитивним (наприклад, відношення “дорівнювати”). Для відношення ж толерантності властивість транзитивності в цілому не виконується (наприклад, відношення “бути схожим”).

Набори інформативних та інваріантних ознак можуть збігатися, але це зовсім не обов'язково.

Інколи, щоб уникнути непорозумінь, ми називатимемо об'єкти і класи Реального світу відповідно Р-об'єктами і Р-класами. Зрозуміло, що Р-об'єкт може належати будь-якій кількості Р-класів.

Р-об'єкти часто називаються реалізаціями, або зразками Р-класів.

Можна виокремити такі основні властивості класів:

- Усі представники класу мають певний набір спільних ознак (впливає з визначення).
- Змінюваність реалізацій класів. По-перше, різні об'єкти, що належать одному класу, можуть бути не схожими між собою. Вони повинні мати спільні інваріантні ознаки, але всі інші ознаки можуть як завгодно варіювати. По-друге, один і той самий об'єкт може змінюватися з часом і навіть поступово переходити від одного класу до іншого (наприклад, перетворення пуголовка на жабу). Усе це свідчить про те, що розпізнати чіткі межі класу часто неможливо.
- Ознайомлення з деякою скінченою кількістю представників одного класу дає можливість впізнавати інших представників цього класу.

Взагалі кажучи, ця властивість може не виконуватися. Але якщо ця властивість виконується (принаймні, теоретично), це є дуже сильним і важливим твердженням. Воно по суті означає можливість навчатися на прикладах, тобто на основі спостереження певної кількості прикладів (можливо, разом з контрприкладми), сформулювати правило розпізнавання, яке дає змогу відрізнити представників даного класу від представників іншого (можливо, з певною достовірністю тобто з певним процентом помилок). У деяких випадках правилом розпізнавання може бути предикат, який залежить від інформативних або інваріантних ознак, інколи правило розпізнавання реалізується у вигляді деякої складної процедури.

Якщо навчання на прикладах неможливе або неефективне, правило розпізнавання інколи можна задати явно. Якщо це зробити не вдається, єдиною можливістю для надійного розпізнавання залишається запам'ятовування всіх можливих представників даного класу. Цей випадок не становить інтересу з теоретичної точки зору, і його можна

реалізувати лише, якщо кількість можливих представників не є надто великою.

Створення пристроїв, що виконують функції розпізнавання різних об'єктів, у багатьох випадках відкриває можливість заміни людини як елемента складної системи спеціалізованим автоматом. Така заміна дозволяє значно розширити можливості різних систем, що виконують складні інформаційнологічні задачі. Якість робіт, що виконує людина на будь-якому робочому місці залежить від кваліфікації, досвіду, сумлінності, стану. У той же час автомат, що її замінює, діє одноманітно і забезпечує завжди однакову якість, якщо він справний.

Але не тільки зазначена заміна і звільнення людини від виконання рутинних операцій є причиною створення і пошуку шляхів створення ряду систем розпізнавання. У деяких випадках людина узагалі не в змозі вирішувати цю задачу зі швидкістю, що задається обставинами, не залежно від якостей і психологічного стану приймаючого рішення (наприклад: протиракетний маневр літака в складних метеоумовах). А автомат з такими задачами може легко справлятися.

Отже, основні цілі заміни людини в задачах розпізнавання зводяться до наступного:

- Звільнення людини від одноманітних рутинних операцій для вирішення інших важливіших задач.

Підвищення якості виконуваних робіт

- Підвищення швидкості вирішення задач.

Практичні реалізації методів розпізнавання носять назву систем розпізнавання (СР).

Центральним завданням розпізнавання образів є побудова на основі систематичних теоретичних і експериментальних досліджень ефективних обчислювальних засобів (об'єднаних в понятті "Системи розпізнавання") для віднесення описів з об'єктів, явищ, процесів до відповідних класів.

Таким чином, СР – складна динамічна система, яка складається в загальному випадку з колективу підготовлених фахівців і сукупності технічних засобів здобуття і переробки інформації, що забезпечують на основі спеціально сконструйованих алгоритмів рішення задачі класифікації відповідних об'єктів, явищ або процесів.

Постають такі запитання:

- які характеристики ліній передачі даних від рецепторів до мозку як ЦВС;
- які ознаки виділяє система обробки;
- які алгоритми використовує мозок для вирішення завдання класифікації, оптимального управління процесом розпізнавання;
- як людині вдається позбавитися від специфічності, властивій технічним СР і тому подібні.

Отже, розпізнавання образів в техніці – необхідний елемент процесу механізації і автоматизації машин, пристроїв і систем для

- заміни людини там, де використовується важка фізична праця;
- реалізації швидких реакцій в управлінні там, де немає часу на роздуми;
- заміни людини в так званих рутинних операціях, тобто, діях, що повторюються, і не вимагають розумових зусиль.

В результаті зіставлення конкретних рішень і розробок виявилось, що не дивлячись на різноманіття і особливості додатків, завдання створення систем розпізнавання мали багато загального, не залежного від вказаної специфіки.

Ось чому для вироблення методичних підходів теорії розпізнавання мало сенс виділяти загальні прийоми, що повторюються, а їх число природно має бути обмеженим і легко об'єднуваним в завдання. Самі ж ці завдання повинні бути ключовими для створення будь-якої системи розпізнавання. В результаті виявилось, що знайдений методичний підхід

до побудови систем розпізнавання образів інваріантний до предметної області.

На прикладах видно, що підходи до побудови систем розпізнавання практично нічим не відрізняються, не дивлячись на специфіку самих створюваних систем.

В результаті ми отримали загальні уявлення про послідовність вирішення і складові завдання створення системи розпізнавання. Не дивлячись на відмінність предметних областей підходи до побудови СР – однакові. Система розпізнавань захворювань серця будувалася так само, як і система розпізнавання літаків, але замінити її вона не може. Аналогічно СР літаків не може застосовуватися для вирішення завдань розпізнавання захворювань серця.

Системи розпізнавання об'єктів (явищ), що створюються людиною завжди вузько спеціалізовані на відміну від її власних природних можливостей.

Що ж до загального підходу до побудови будь-якої системи, то тепер, якщо у нас є деяка сукупність об'єктів або явищ, які необхідно розпізнавати (класифікувати), на основі узагальнення дій при створенні СР в двох розглянутих прикладах ми знаємо, що послідовність вирішення відповідних завдань наступна:

1. відповідно до вибраного принципу сукупність об'єктів або явищ підрозділяється на ряд класів (говорять: призначається алфавіт класів);
2. розроблюється сукупність ознак (говорять: словник);
3. на мові словника ознак описується кожен клас; -вибираються і (або) створюються засоби визначення ознак;
4. на обчислювальних засобах реалізується алгоритм зіставлення апостеріорних і апріорних даних і приймається рішення про результати розпізнавання.

Хоча послідовність дій визначена, залишаються запитання:

1. як краще здійснювати розбиття об'єктів (літаки, захворювання і ін.) на класи;
2. як накопичувати і обробляти апріорну інформацію;
3. із яких міркувань вибирати ознаки; -як описувати класи на мові ознак;
4. на основі яких методів порівнювати апріорну і апостеріорну інформацію;
5. коли і як з'являється вся система розпізнавання.

Отже, головні висновки:

- Завдання, що вирішуються в процесі створення систем розпізнавання, інваріантні відносно предметної області, мають багато спільного, базуються на єдиному методологічному підході.
- Кожна система розпізнавання індивідуальна і призначається лише для одного цілком конкретного виду об'єктів або явищ. Якщо знайдена сфера застосування розпізнавання, то відповідна система повинна розроблятися заново з урахуванням нових специфічних властивостей об'єктів (явищ), що визначають як систему вимірів характеристик, так і словник ознак, алфавіт класів і алгоритм ухвалення рішень.
- СР повинна створюватися методом послідовних наближень внутрішньої структури на її математичній моделі у міру накопичення необхідної інформації.

Метод січних площин – це алгоритм навчання, заснований на методі січних площин, полягає у відокремленні образів за допомогою частин гіперплощин.

Алгоритм складається з наступних етапів:

Навчання (формування гіперплощин для відокремлення множин):
 проведення січних площин
 вилучення зайвих гіперплощин

а. вилучення зайвих частин площини

2. Розпізнавання нових об'єктів

Припустимо, що потрібно навчити комп'ютер розпізнавати 3 образи а, б, та с. Алгоритм матиме наступний вигляд.

У комп'ютер вводять коди двох точок, які належать різним образам та проводять довільну пряму, яка їх відокремлює (рис. 2.1).

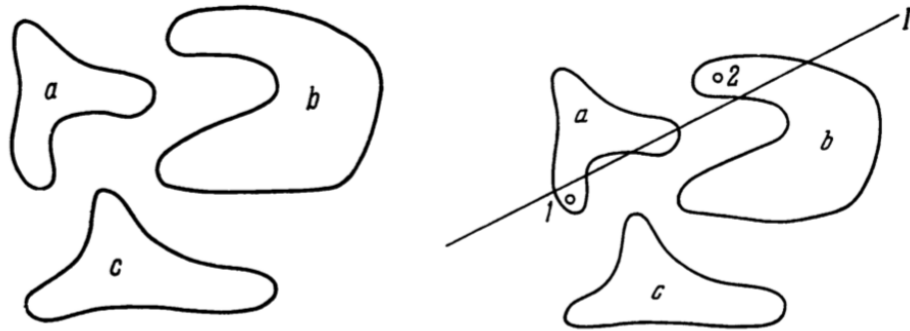


Рисунок 2.1 – Проведення довільної прямої через 3 об'єкти

Далі береться третій об'єкт і перевіряється правильність класифікації відносно проведеної прямої (рис. 2.2).

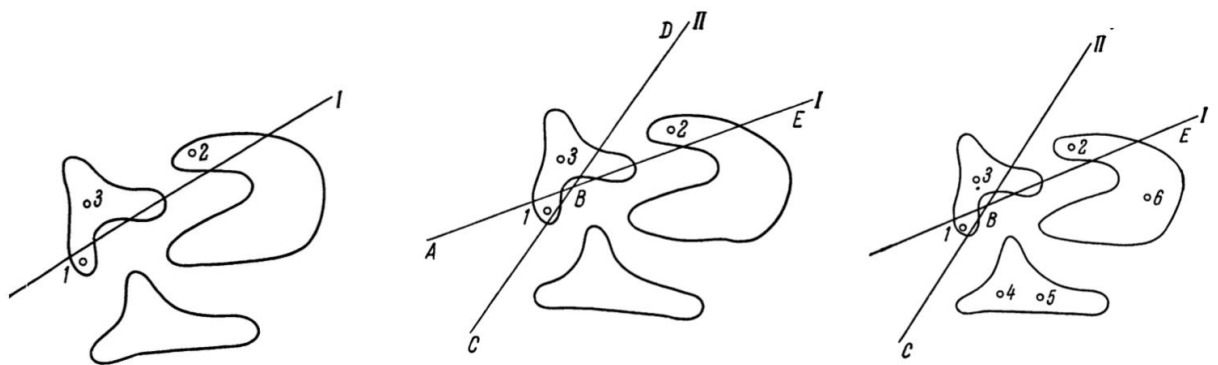


Рисунок 2.2 – Перевірка правильності класифікації відносно прямої

Оскільки третя точка належить тій самій півплощині, що й друга, то приходимо до висновку, що поточний класифікатор працює невірно. Для відокремлення точок 2 та 3 проводиться пряма II. Після цього площина

розбивається на 4 частини. Ті частини, у який лежать об'єкти 1 та 3 (області ABC та ABD) відносимо до класу а, область DBE — до класу b.

При перевірці нової точки можливі три випадки:

1. виникає протиріччя (як це було для точки 3)
2. протиріччя не виникає, точка потрапляє у свою частину простору
3. протиріччя не виникає, точка потрапляє у вільну (непозначену) частину простору (точки 4 і 5)

У третьому випадку (точки 4 та 5 на рис. 2.2) машина відносить вільну частину площини до відповідного образу.

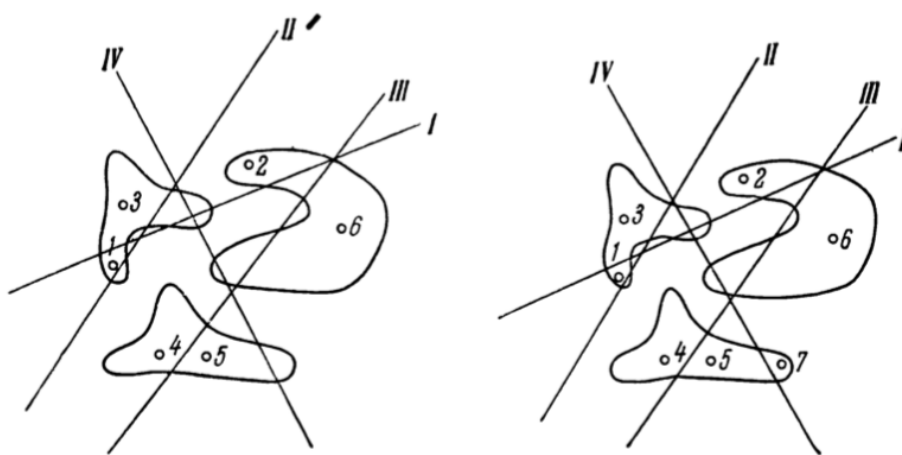


Рисунок 2.3 – Проведення двох нових прямих

Для відокремлення точки 6 (образ b) від точок 4 та 5 (образ c) потрібно провести дві нові прямі (прямі III та IV на рис. 2.3).

Подальші кроки вищенаведеного процесу навчання наведені на рис. 2.3., де вказано проміжкові та кінцевий результат стадії а). Незафарбовані ділянки не відносяться розпізнавальною системою до жодного із трьох наведених образів.

З рис. 2.4 видно, що гіперплощини (прямі) I, III та V можуть бути вилученні цілком, оскільки вони не містять частин, відкидання яких призводить до появи протиріч. Наприклад, пряма I використовується лише для відокремлення об'єктів 1 та 2 — представників різних образів а та b.

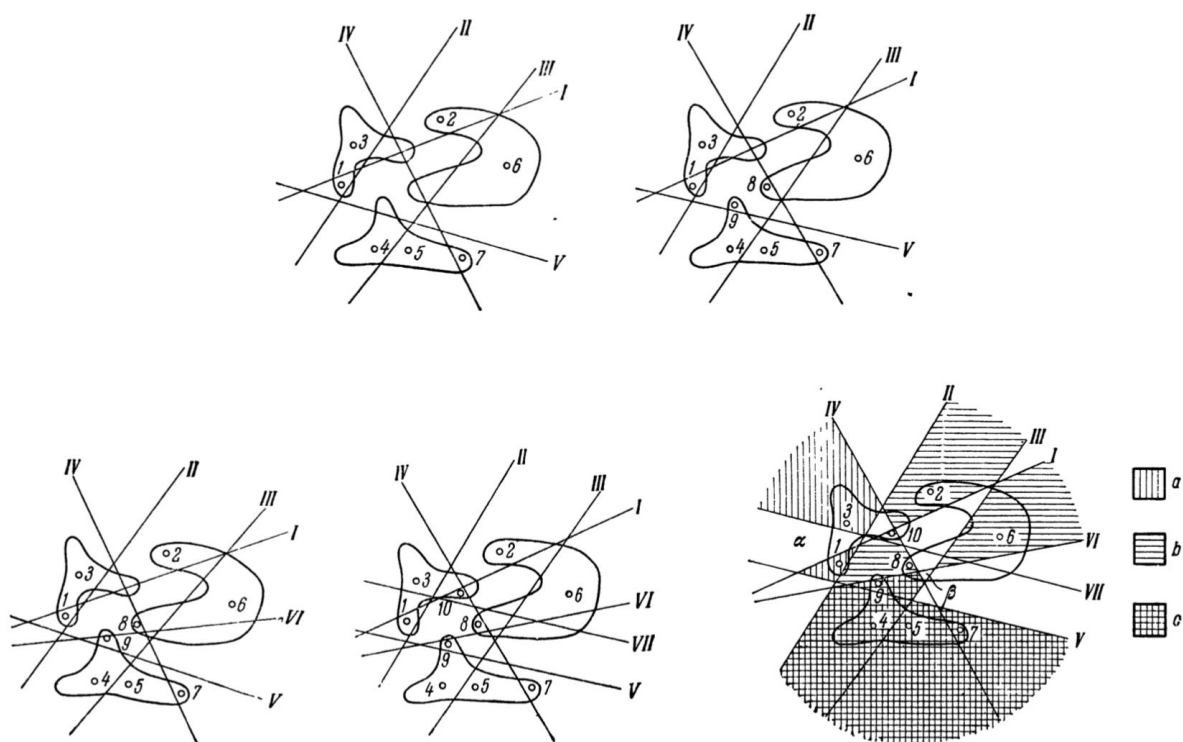


Рисунок 2.4 – Гіперплощини I, III та V не містять частин, відкидання яких призводить до появи протиріч

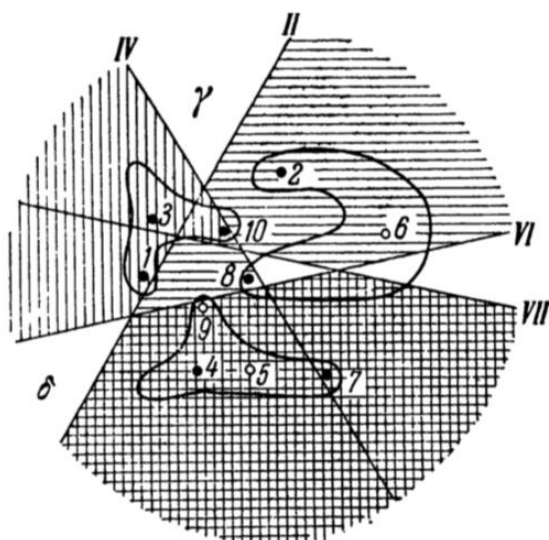


Рисунок 2.5 – Розбиття після відкидання зайвих площин

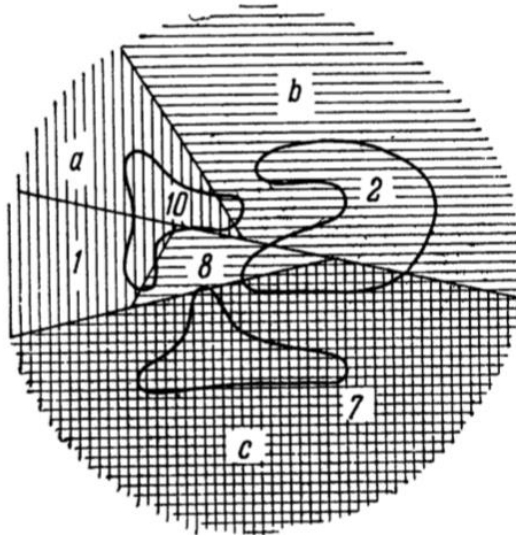


Рисунок 2.6 – Кінцевий результат

Але ці об'єкти відокремлює також пряма IV (або пряма VI). Розбиття після відкидання зайвих площин наведено на рис. 2.5. Слід зазначити, що у результаті частка вільних частин площини зменшилися. Тому спочатку виконуємо перевірку усіх частин площини II, потім усіх частин площини IV і т.д. Кінцевий результат наведений на рис 2.6. Математично опишемо алгоритм. Випадковим чином вибираємо n чисел $\lambda_1, \dots, \lambda_n$. Далі для координат відповідних точок обчислюємо значення сум:

$$\sigma^{(1)} = \sum \lambda_i x_i^{(1)}$$

$$\sigma^{(2)} = \sum \lambda_i x_i^{(2)},$$

Після цього вибирається число $\lambda_{n+1} = (\sigma^{(1)} + \sigma^{(2)}) / 2$ (вважаємо, що значення сум різні). Тоді рівняння

$$\lambda_1 x_1 + \dots + \lambda_n x_n - \lambda_{n+1} = 0$$

Будемо позначати положення точки відносно площини знаком відхилення $\lambda(x) = \lambda_1 x_1 + \dots + \lambda_n x_n - \lambda_{n+1}$. Далі процес описується за допомогою таблиць знаків. Кінцевий результат кроку а. наведений у Таблиці 2.1.

Таблица 2.1

Таблиця знаків								
Номер точки	Образ	Номер площини						
		I	II	III	IV	V	VI	VII
		Знак точки						
1	a	0	1	1	1	0	1	1
2	b	1	0	1	0	0	1	0
3	a	1	1	1	1	0	1	0
4	c	0	0	1	1	1	0	1
5	c	0	0	0	1	1	0	1
6	b	0	0	0	0	0	1	0
7	c	0	0	0	0	1	0	1
8	b	0	0	1	1	0	1	1
9	c	0	0	1	1	0	0	1
10	a	0	0	1	1	0	1	0

Далі відбувається перевірка можливості видалення стовпців. Виконується перевірка того, чи залишаються унікальними рядки після відкидання стовпчика. Якщо так, то відповідну площину можна відкинути. Результат зображений у Таблиці 2.2. У цій таблиці є однакові рядки знаків. Зайві точки також можна виключити. Результат наведений у останній таблиці.

Таблиця 2.2

Таблиця знаків					
Номер	Образ	Номер площини			
		II	IV	VI	VII

точки		Знак точки			
1	a	1	1	1	1
2	b	0	0	1	0
3	a	1	1	1	0
4	c	0	1	0	1
5	c	0	1	0	1
6	b	0	0	1	0
7	c	0	0	0	1
8	b	0	1	1	1
9	c	0	1	0	1
10	a	0	1	1	0

Розпізнавання нових об'єктів. Обчислюються знаки об'єкту відносно усі гіперплощин і отриманий код порівнюється з усіма рядками таблиці знаків. При співпаданні рядків об'єкт відноситься до відповідного образу.

Розглянемо метод потенціалів для розпізнавання образів. Точковий електричний заряд у однорідному середовищі створює електричне поле, зображене на рис. 2.7. Радіальні лінії — це силові лінії поля, концентричні кола — лінії однакового потенціалу.

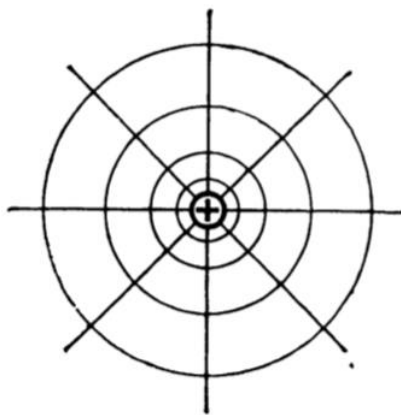


Рисунок 2.7 – електричне поле

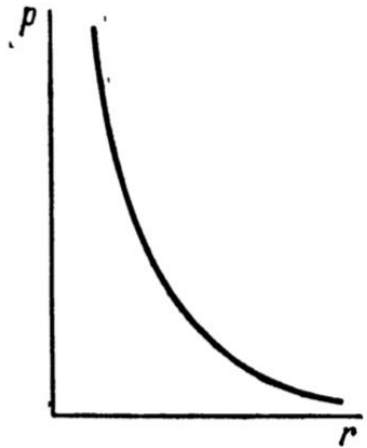


Рисунок 2.8 – Крива зміни потенціалу як функції відстані

Потенціал p у кожній точці простору визначається співвідношенням

$$p = a \frac{q}{r^2},$$

де a — деяка стала, q — величина заряду, r — відстань від заданої точки до заряду.

Крива зміни потенціалу як функції відстані наведена на рис. 2. 8. Потенціал зменшується по мірі віддалення від його джерела. Якщо поле утворене кількома зарядами, то потенціал в кожній точці рівний сумі потенціалів, які створюються кожним із зарядів.

Припустимо, що у просторі розташовані дві компактні групи зарядів. У одній групі – позитивні, у другій – негативні. На рис. 2.9 показаний розподіл потенціалів у околі цих зарядів, на рис. 2.10 ці потенціали алгебраїчно просумовані.

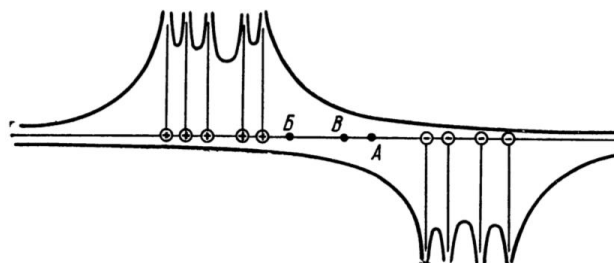


Рисунок 2.9 – Розподіл потенціалів у околі зарядів

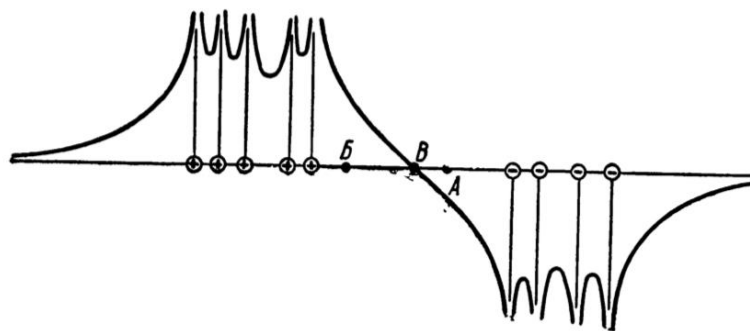


Рисунок 2.10 – Алгебраїчно просумовані потенціали у околі зарядів

Точку можна віднести до тієї чи іншої множини точок у залежності від того, який знак має сумарний потенціал поля у цій точці.

Вище наведені міркування по аналогії можна перенести на точки простору рецепторів. Кожній точці, яка з'являється у процесі навчання, поставимо у відповідність деяку функцію, аналогічну по формі до електричного потенціалу. Такою функцією може бути, наприклад, функція

$$\varphi(D) = \frac{1}{1 + \alpha D^2},$$

де α — деякий коефіцієнт, D — відстань (у деякій метриці) між точкою-джерелом потенціалу та точкою, у якій обчислюється потенціал. Наприклад, у якості D можна використовувати евклідову відстань, віддаль Хеммінга, манхеттенську метрику, тощо.

Нехай джерелами потенціалів є група точок, які відповідають деякому образу a . Тоді можна вважати, що середній потенціал, які створюють у деякій точці простору джерела цього образу, характеризує віддаль від цієї точки до усього образу у цілому.

Спираючись на гіпотезу компактності, можна запропонувати наступне правило розпізнавання: точку відносимо до того образу, середній потенціал якого у цій точці є максимальним.

Розглянемо алгоритм розпізнавання на основі методу потенціалів.

Важливим є те, що у процесі навчання запам'ятовуються координати усіх точок та належність їх до відповідних образів a_1, \dots, a_m . Наступним кроком є саме розпізнавання.

Для точки x , яка підлягає розпізнаванню, обчислюється потенціали кожного образу, тобто суми

$$\Phi_i(x) = \frac{1}{n_i} \sum_{a \in a_i} \varphi_a(x), \quad i = 1, 2, \dots, m,$$

де m — кількість різних образів, n_i — кількість точок відповідного образу, використаних у процесі навчання,

$$\varphi_a(x) = \frac{1}{1 + \alpha D^2(a, x)}$$

– потенціал, який створює точка a у точці x .

Порівнюються величини $\Phi_1(x)$, $\Phi_2(x)$, ..., $\Phi_m(x)$ і точка x відноситься до того образу, потенціал якого є найбільшим.

У випадку двох образів a_1 та a_2 рішення можна приймати на основі значення знаку функції

$$\Delta\Phi(x) = \Phi_{a_1}(x) - \Phi_{a_2}(x).$$

Застосування вищенаведеного алгоритму для розпізнавання цифр, зображених за допомогою рецепторного поля 610 у середньому дало змогу досягнути правильного розпізнавання у 85% випадків за умови, що $n_1 = n_2 = \dots = n_{10} = 12$, причому подальше збільшення обсягу навчальної вибірки практично не впливає на якість розпізнавання. Недоліки найпростішого алгоритму виявляються у тому випадку, коли точки навчальної вибірки розподілені нерівномірно. Відповідний приклад наведений на Рис. 18-20.

Навчальна вибірка для образів a та b , зображених на рис. 2.11, наведена на рис. 2.12. У цьому випадку у зв'язку із «невдалою» навчальною вибіркою, нова точка 2 (рис. 2.13) буде віднесена до образу b . Більш того, машина може не впізнати навіть відомий їй об'єкт, який

зустрічався у процесі навчання, наприклад, точку 1, яка належить до образу а.

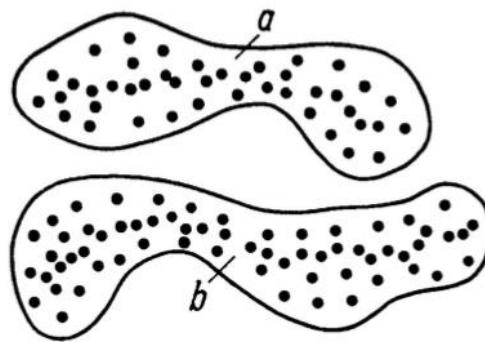


Рисунок 2.11 – Набір образів

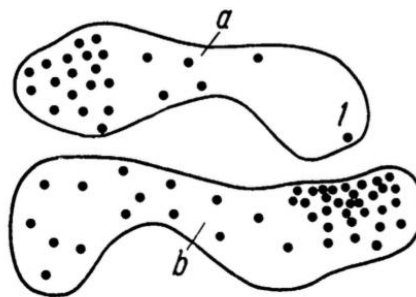


Рисунок 2. 12 – Навчальна вибірка для образів а і b

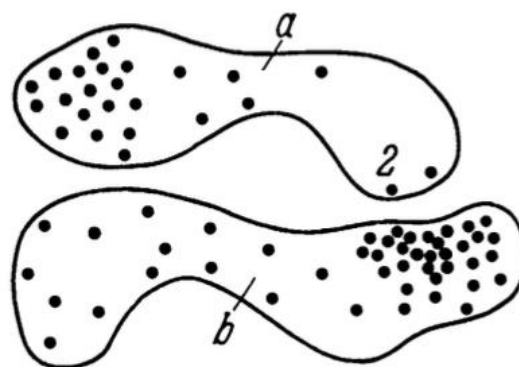


Рисунок 2.13 – Точка 2 буде віднесена до образу b

Алгоритм розпізнавання може бути покращений, шляхом введення поняття «ваги точки» та доповненням кроку навчання наступною операцією. Після за- пам'ятовування усіх об'єктів їм присвоюють

початкову вагу 1. Далі до елементів навчальної вибірки застосовують крок 2 алгоритму навчання, причому потенціали середні потенціали обчислюють за формулою

$$\Phi_i(\mathbf{x}) = \frac{1}{n_i} \sum_{\mathbf{a} \in a_i} \varphi_{\mathbf{a}}(\mathbf{x}) \cdot w(\mathbf{a}), \quad i = 1, 2, \dots, m,$$

де $w(\mathbf{a})$ — вага точки \mathbf{a} .

Якщо при розпізнаванні об'єкту \mathbf{a} відбувається помилка, то вагу $w(\mathbf{a})$ відповідного об'єкту збільшують на деяку величину, наприклад на одиницю. Потім застосовується ще один такий самий цикл перевірки та корекції ваг і т.д. Цикли повторюються до тих пір, поки усі відомі фігури не будуть розпізнані правильно.

Застосування покращеного алгоритму до розпізнавання дало змогу підвищити відсоток правильного розпізнавання нових зображень цифр до 89%.

У випадку бінарних зображень традиційні функції відстані Хеммінга або Евкліда не завжди точно відображають відмінність між зображеннями. Відповідний приклад наведений на Рис. 21. Шляхом зсуву вертикальної лінії із п'ятірки (Рис. 2.14, а), можна отримати «трохи гіршу» п'ятірку (рис. 2.14, б) та трійку (рис. 2.14, в), кожна з яких відрізняється від початкової фігури у шести розрядах. Тобто, елемент «свого» образу та елемент «чужого» образу знаходяться на однаковій відстані від початкової фігури. Схожий приклад наведений на рис. 2.15 для випадку зображень одиниць та двійок, які відрізняються у десяти розрядах.

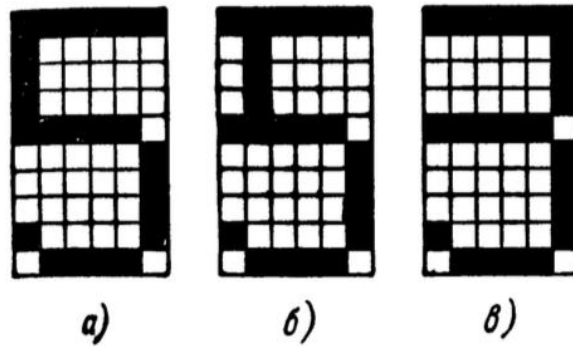


Рисунок 2.14 – Відмінність у зображеннях при зсуві. П'ятірка стає трійкою

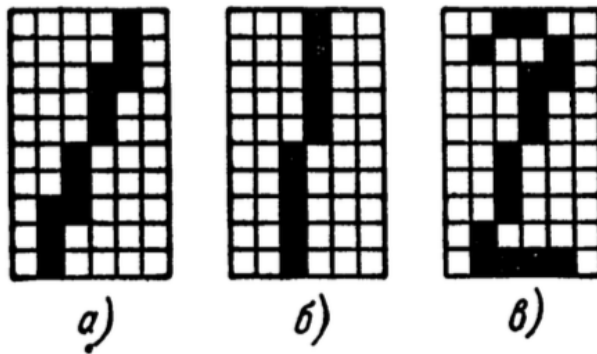


Рисунок 2. 15 – Відмінність у зображенні при зсуві. Одиниця стає двійкою

Ці приклади показують, що значні зміни елементів зображень, які значно спотворюють зображення, приводять тих самих змін у відстані, що і незначні. Тому застосований метод кодування недостатньо добре передає відмінність фігур.

Надійність розпізнавання може бути підвищена шляхом удосконалення методу кодування. Поставимо у відповідність кожному рецептору функцію, яка рівна одиниці на цьому рецепторі, спадаючу по усім напрямкам від нього, тобто функцію, аналогічну до потенціалу s (рис. 2. 16, а) . Ця функція може бути апроксимована ступінчастими функціями (рис. 2. 16, б-в).

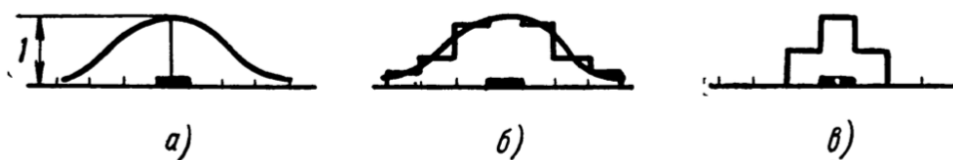


Рисунок 2.16 – Апроксимована ступінчастими функціями

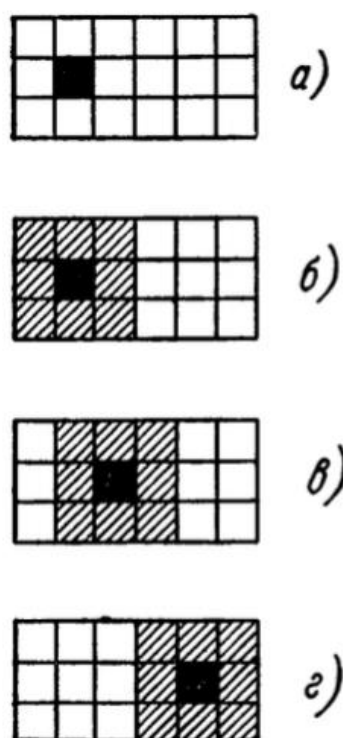


Рисунок 2.17 – Груба апроксимація

У випадку «грубої апроксимації» функція рівна одиниці на відповідному елементів рецепторного поля, $1/2$ — на сусідніх елементах, нулю — на всіх інших. Для прикладу розглянемо одно кліткову фігуру, зображену на рис. 2.17, а. На рис. 2.17, б зображено значення ступінчастої функції відстані. Код такої фігури має вигляд

$$\frac{1}{2} \frac{1}{2} \frac{1}{2} 000 \frac{1}{2} 1 \frac{1}{2} 000 \frac{1}{2} \frac{1}{2} \frac{1}{2} 000.$$

Посунемо фігуру на один елемент управо (рис. 24, в). Отримаємо фігуру із кодом

$$0 \frac{1}{2} \frac{1}{2} \frac{1}{2} 000 \frac{1}{2} 1 \frac{1}{2} 000 \frac{1}{2} \frac{1}{2} \frac{1}{2} 00.$$

Евклідова відстань між цими фігурами рівна $\sqrt{2}$. Тепер посунемо фігуру іще на дві клітинки вправо. Отримаємо фігуру (Рис. 24, в) із кодом

$$0 \frac{1}{2} \frac{1}{2} \frac{1}{2} 000 \frac{1}{2} 1 \frac{1}{2} 000 \frac{1}{2} \frac{1}{2} \frac{1}{2} 00.$$

Відстань від цієї фігури до початкової рівна $\sqrt{6}$. Як видно із прикладів, при такому методі кодування «більш віддаленим» фігурам відповідає більша відстань у просторі рецепторів.

Описаний вище прийом можна застосувати для більш складних фігур з використанням сумарних потенціалів рецепторів. На рис. 2.18 зображено кодування цифр із прикладу, наведеного на рис. 2.14.

2	2,5	2	2	2	1,5
2,5	2,5	1,5	1,5	1,5	1
2	1,5	0	0	0	0
2,5	2,5	1,5	1,5	1	0,5
2	2,5	2	2	2	1
1	1,5	1,5	1,5	2	2
0	0	0	0	1,5	2
0,5	0,5	0	0	1,5	2
1,5	1,5	1,5	1,5	2	2
1	2	2	2	2	1

2	2,5	2,5	2	2	1,5
2	3	2,5	1,5	1,5	1
1,5	2	1,5	0	0	0
2	3	2,5	1,5	1	0,5
2	2,5	2,5	2	2	1
1	1,5	1,5	1,5	2	2
0	0	0	0	1,5	2
0,5	0,5	0	0	1,5	2
1,5	1,5	1,5	1,5	2	2
1	2	2	2	2	1

1,5	2	2	2	2,5	2
1	1,5	1,5	1,5	2,5	2,5
0	0	0	0	1,5	2
1	1,5	1,5	1,5	2	2
1,5	2	2	2	2	1,5
1	1,5	1,5	1,5	2	2
0	0	0	0	1,5	2
0,5	0,5	0	0	1,5	2
1,5	1,5	1,5	1,5	2	2
1	2	2	2	2	1

Рисунок 2.18 – кодування цифр

Тоді відстань між «п'ятірками» рівна 2,25, а відстань між першою п'ятіркою та трійкою — 5,22. Застосування удосконаленого методу кодування дозволило збільшити середню надійність розпізнавання букв до 94%.

2.3 Визначення перспективних нейромережових архітектур для розпізнавання тексту на зображеннях

Проведений аналіз сучасного стану найромережових технологій дозволяє сформулювати висновок про те, що доцільність застосування конкретного типу НМ слід визначати на основі співставлення

характеристик мережі з умовами прикладної задачі. До вказаних характеристик та умов відносяться:

- параметри навчальних даних,
- загальні обмеження процесу навчання,
- вимоги до обчислювальних потужностей,
- вимоги до вихідної інформації, обмеження технічної реалізації НМ,
- сфера застосування.

Розглянемо вказані характеристики в ракурсі комп'ютерної системи розпізнавання голосових сигналів.

До основних параметрів навчальних даних відносяться:

- Кількість параметрів, що характеризують навчальний приклад.
- Вид параметрів, дискретний s (символьний) чи безперервний (числовий).
- Загальна кількість навчальних прикладів.
- Наявність помилок (шуму) в навчальних прикладах.
- Наявність кореляції навчальних прикладів.
- Можливість та необхідність попередньої обробки вхідних даних з метою їх нормалізації та видалення шуму.
- Повнота вибірки, тобто можливість відображення в ній всіх аспектів процесу, що моделюється.

Пропорційність навчальних прикладів, що відповідають різним аспектам процесу, що моделюється.

Загальні обмеження процесу навчання обумовлюються:

- Максимальним терміном навчання.
- Необхідністю представлення в навчальних даних очікуваного вихідного сигналу НМ. Цим визначається тип навчання – з вчителем або без вчителя.

- Можливістю автоматизації процесу навчання, яка визначається кількістю та важливістю емпіричних параметрів. Вказана можливість багато в чому визначає умови застосування НМ. Мережі в яких процес навчання не автоматизовано можуть використовуватись тільки в лабораторних умовах.
- Можливістю донавчання в процесі експлуатації.
- Вимогами до якості навчання, яке звичайно оцінюють по величині максимальної та середньої помилки розпізнавання навчальних та тестових даних. При цьому тестові дані повинні не значно відрізнятись від навчальних.
- Можливістю навчання НМ в лабораторних умовах. Доцільність навчання в лабораторних умовах пояснюється потребами оптимального механізму створення та оновлення бази знань НМ.

На практиці вимоги до обчислювальних потужностей визначаються максимальною кількістю прикладів (обсяг пам'яті), яку може запам'ятати мережа для досягнення необхідної достовірності прийняття рішення. В свою чергу достовірність прийняття рішення характеризується допустимими величинами максимальної та середньої помилки мережі на реальних даних які в загальному випадку можуть виходити за межі множини навчальних даних. Відповідно виникає задача екстраполяції результатів навчання НМ за межі навчальних прикладів. Відзначимо, що обчислювальна потужність мережі залежить від її типу та алгоритму навчання. Ще однією вимогою може бути незмінність виходу мережі для різних прикладів з однаковими параметрами.

Вимоги до вихідної інформації НМ вказують на те в якому вигляді має бути представлена ця інформація. Наприклад, при розпізнаванні слів може виникнути необхідність не тільки визначення ситуації “слово А

присутнє”, але й розрахунку ймовірності появи цієї ситуації. Також вимогою може бути необхідність визначення вербальних залежностей між вхідною та вихідною інформацією.

Обмеження технічної реалізації НМ стосуються: швидкості прийняття рішення, інтеграції в існуючі ЗЗІ, обсягу та складності програмної реалізації. Для зменшення обсягу можливо розділити програмний код для навчання мережі від коду, що відповідає за її функціонування.

Сфера застосування визначає ЗЗІ в яких буде використовуватись НМ. На сьогодні достатньо дослідженим є використання НМ для розпізнавання образів та при проведенні оптимізаційних розрахунків. Відзначимо, що системи розпізнавання образів принципово відрізняються від систем аналізу тексту тим, що в них кількість вихідних та кількість комбінацій вхідних параметрів принципово обмежена. В системах аналізу тексту ця кількість принципово необмежена. В перспективі доцільно застосувати НМ з метою реалізації паралельних розрахунків в КС, що дозволить значно підвищити їх стійкість від багатьох типів атак з метою відмови в обслуговуванні.

Крім того сфера застосування визначається пристосованістю мережі до автономного функціонування. Для цього в архітектурі НМ повинно бути передбачено можливість повної автоматизації процесу донавчання на експлуатації.

Якісні оцінки відповідності основних характеристик НМ умовам задач захисту ПЗ для описаних в п. 1.1-1. 8 типів мереж наведені в табл. 2. 1. В табл. 2.1 відсутні характеристики, які хоча і застосовуються при побудові мережі, але не впливають на вибір типу НМ. Оцінки відповідності виставлені в числовому вигляді по трьохбальній системі (-1 – мінімальна, 0 – середня, 1 – максимальна). Відсутність оцінки означає, що для її визначення потрібні додаткові дослідження.

Використання даних Таблиці 3 дозволяє визначити принципову

доцільність застосування того чи іншого типу НМ для вирішення задачі. Остаточне рішення про використання конкретного типу НМ із декількох можливих повинно бути прийняте після проведення порівняльних експериментів.

Таблиця 2.3

Якісні оцінки відповідності НМ умовам задач розпізнавання тексту

Умова	БШП	РБФ	ОМ	АР Т	СН М	PNN/ GRN N	Асоціа -тивні
sНавчальні дані							
sДопустимість шуму	1	0	1	-1	1	0	-1
Допустимість кореляції	1	1	1	1	1	1	-1
Повнота виборки	-1	1	1	-1	-1	1	0
Пропорційність прикладів	1	-1	-1	-1	-1	-1	0
Загальні обмеження процесу навчання							
Короткий термін навчання	-1	0	1	1	0	1	1
Представлення в навчальних прикладах очікуваного виходу	1	1	-1	-1	-1	1	1
sАвтоматизація навчання	1	-1	0	1	1	1	0
Можливість донавчання	0	1	1	1	1	1	0
sЯкість навчання	1	0	0	1	1	1	1
Обчислювальні потужності							
Обсяг пам'яті	1	-1	-1	-1		-1	0
Екстраполяції результатів навчання	1	-1	-1	-1		-1	1
Незмінність результатів	1	1	0	1	1	1	0
Вихідна інформація							
Інтерпретації виходу у вигляді ймовірності	0	0	-1	-1	-1	1	0
Інтерпретації виходу у графічному вигляді	-1	-1	1	-1	-1	-1	-1
Можливість вербалізації	1	0	-1	-1	-1	0	-1
Обмеження технічної реалізації НМ							
Швидкості прийняття рішення	1	1	1	1	0	1	-1
sОбсяг програмної реалізації	-1	1	-1	0	-1	-1	0
Сфера застосування							
Системи розпізнавання образів	1	1	1	1	0	1	1

Системи аналізу тексту	-1	-1	1	0	1	0	-1
Системи управління	-1	-1	1	-1	-1	-1	1
Автономність функціонування	-1	-1	-1	1	1	-1	-1

Відповідно матеріалів даної роботи та результатів можна зробити висновок про те, що основними напрямками застосування НМ в галузі комп'ютерного забезпечення технічних та економічних систем є розпізнавання образів, визначення оптимальних управляючих рішень та створення асоціативної пам'яті. До першого напрямку віднесемо задачі класифікації образів, кластеризації образів та апроксимації функцій.

Зазначимо, що до групи задач апроксимації функції слід віднести розрахунок параметрів процесів, що відбуваються в технічних системах.

Адже по своїй суті оцінка регресивних або прогнозованих значень параметрів деякого процесу є апроксимацією функції, що описує цей процес. До другого напрямку віднесемо власне задачі оптимального управління та задачі управління з еталонною моделлю.

До третього напрямку входять задачі створення інформаційно-обчислювальних систем з пам'яттю, що адресується за змістом.

Для вирішення поставленого завдання найдоцільніше застосовувати рекурентні нейронні мережі, які в процесі роботи можуть зберігати інформацію про своїх попередніх станах. Далі ми розглянемо принципи роботи таких мереж на прикладі рекуррентной мережі Елмана.

2.4 Використання рекурентних нейронних мереж

Штучна нейронна мережа Елмана, відома так само як `imple Recurrent Neural Network`, складається з трьох шарів — вхідного (розподільного) шару, прихованого і вихідного (обробних) шарів. При цьому прихований шар має зворотний зв'язок сам на себе. На рис. 2.19 представлена схема нейронної мережі Елмана.

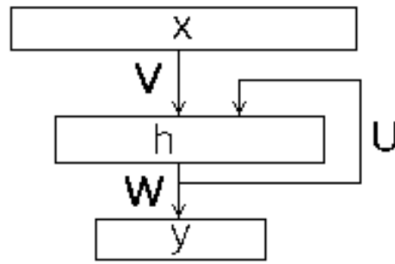


Рисунок 2.19 – Схема нейронної мережі Елмана

На відміну від звичайної мережі прямого поширення, вхідний образ рекурентної мережі — це не один вектор, але послідовність векторів. Вектори вхідного образу в заданому порядку подаються на вхід, при цьому новий стан прихованого шару залежить від його попередніх станів. Мережа Елмана можна описати наступними співвідношеннями:

$$h(t) = f(V * x(t) + U * h(t - 1) + b_h)$$

$$y(t) = g(W * h(t) + b_y)$$

де

- $x(t)$ – вхідний вектор номер t ;
- $h(t)$ – стан прихованого слою для входу $x(t)$ ($h_0 = 0$);
- $y(t)$ – вихід мережі для входу $x(t)$;
- U – вагова матриця розподільного слою;
- W – вагова (квадратна) матриця зворотніх зв'язків прихованого слою;
- b_h – вектор зсувів прихованого слою;
- V – вагова матриця вихідного слою;
- b_y – вектор зсувів вихідного слою;
- f – функція активації прихованого слою;
- g – функція активації вихідного слою.

При цьому можливі різні схеми роботи мережі. Залежно від того як сформу вати вхід і вихід рекурентної мережі, можна різними способами задати схему її роботи. Розглянемо цього розгорнемо схему рекурентної мережі в часі (рис. 2.20).

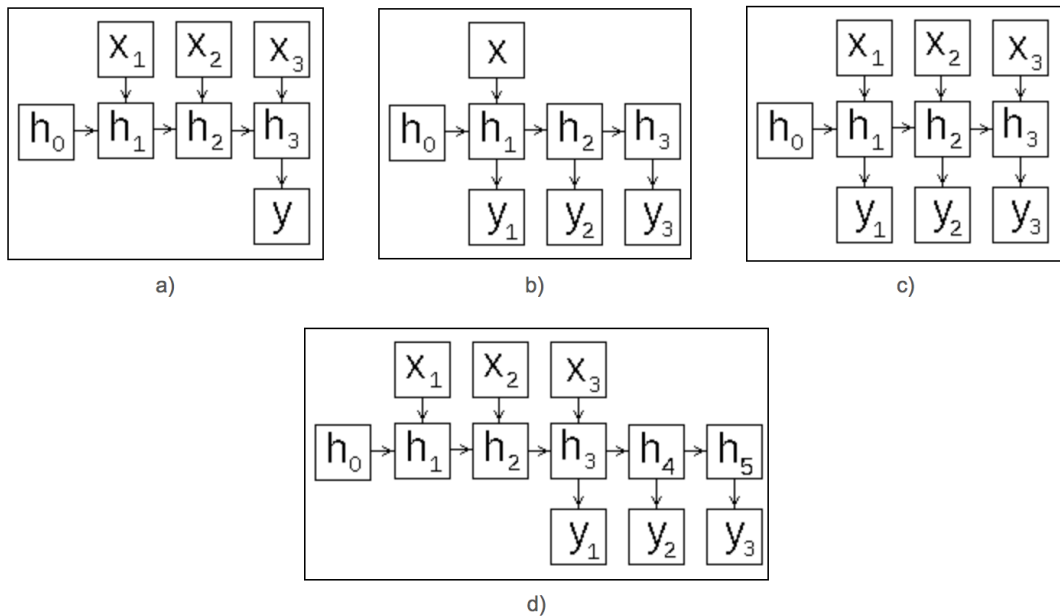


Рисунок 2.20 – Схема роботи рекуррентної нейронної мережі

Існують декілька способів організації роботи рекуррентної мережі:

- "Багато в один" (many-to-one) (рис 2. 20, а) – прихований шар послідовно змінює свій стан, з його кінцевого стану обчислюється вихід мережі, цю схему можна використовувати для класифікації текстів;
- "Один у багато" (one-to-many) (рис 2.20, б) – прихований шар ініціалізується одним входом, з ланцюжка його наступних станів генеруються виходи мережі, цю схему можна використовувати для анотування зображень;
- "Багато в багато" (many-to-many) (рис 2.20, с) – на кожен вхід мережу видає вихід, який залежить від попередніх входів, цю схему можна використовувати для класифікації відео;
- "Багато в багато" (many-to-many) (рис 2. 20, d) – прихований шар послідовно змінює свій стан, його кінцевий стан служить ініціалізацією для видачі ланцюжка результатів, цю схему можна використовувати для створення систем машинного перекладу і чат-ботів.

Розглянемо метод навчання рекуррентної мережі Елмана за

схемою many-to-one (рис. 2.20, а), для реалізації класифікатора об'єктів, заданих послідовностями векторів. Для навчання мережі Елмана застосовуються ті ж градієнтні методи, що і для звичайних мереж прямого поширення, але з певними модифікаціями для коректного обчислення градієнта функції помилки. Він обчислюється за допомогою модифікованого методу зворотного поширення, який носить назву Backpropagation through time (метод зворотного поширення з розгортанням мережі в часі, BPTT). Ідея методу - розгорнути послідовність, перетворивши рекурентну мережу в "звичайну" (рис. 2.20, а). Як і в методі зворотного поширення для мереж прямого поширення, процес обчислення градієнта (зміни ваг) відбувається в три наступних етапи:

- прямий прохід – обчислюємо стану шарів,
зворотний прохід – обчислюємо помилку шарів,
обчислення зміни ваг, на основі даних отриманих на першому і другому етапах.

3. АЛГОРИТМ РОЗПІЗНАВАННЯ АВТОМОБІЛЬНИХ НОМЕРНИХ ЗНАКІВ У СИСТЕМІ КОНТРОЛЬНО ПРОПУСКНОГО ПУНКТУ

3.1 Покращення візуальної якості зображень шляхом поелементного перетворення

Обробка та фільтрація зображень – дуже важлива складова у процесі розпізнавання. Більшість процедур обробки для отримання результату в кожній точці кадру використовує вхідні дані із деякої множини точок вихідного зображення, які оточують точку, що обробляється. Однак є група процедур, де здійснюється так звана поелементно обробка. Тут результат обробки в будь-якій точці кадру залежить тільки від значення вхідного зображення в цій же точці. Очевидною перевагою таких процедур є їх гранична простота. Разом з

тим, багато з них призводять до очевидного суб'єктивного поліпшення візуальної якості. Цим визначається увага, яка приділяється поелементній процедурі. Не перебільшуючи їх ролі, відзначимо, що дуже часто поелементна обробка застосовується як заключний етап при вирішенні більш складної задачі обробки зображення.

Сутність поелементної обробки зображень зводиться до наступного. Нехай $x(i, j) = x_{ij}$, $y(i, j) = y_{ij}$ – значення яскравості вихідного і одержуваного після обробки зображень відповідно в точці кадру, що має декартові координати N (номер строки) і I (номер стовпчика). Поелементна обробка означає, що існує функціональна однозначна залежність між цими яскравостями: $y_{ij} = f_{ij}(x_{ij})$, що дозволяє за значенням вихідного сигналу визначити значення вихідного продукту. У загальному випадку, як це враховано в даному виразі, вид або параметри функції $f_{ij}(x)$, яка описує обробку, залежать від поточних координат. При цьому обробка є неоднорідною. Однак у більшості практично застосовуваних процедур використовується однорідна поелементна обробка. У цьому випадку індекси N і I у виразі можуть бути відсутніми. При цьому залежність між яскравостями вихідного і обробленого зображень описується функцією $y = f(x)$ однаковою для всіх точок кадру.

Розглянемо лінійне контрастування зображення. Завдання контрастування пов'язана з поліпшенням узгодження динамічного діапазону зображення і екрану, на якому виконується візуалізація. Якщо для цифрового представлення кожного відліку зображення відводиться 1 байт (8 біт) пристрою, що запам'ятовує, то вхідний чи вихідний сигнали можуть приймати одне з 256 значень. Звичайно як робочого використовується діапазон 0...255; при цьому значення 0 відповідає при візуалізації рівню чорного, а значення 255 – рівню білого. Припустимо, що мінімальна і максимальна яскравості вихідного зображення рівні p_l і p_k відповідно. Якщо ці параметри або один з них істотно відрізняються

від граничних значень яркостного діапазону, то візуалізована картина виглядає як ненасичена, незручна, втомлюються при спостереженні. Приклад такого невдалого подання наведено на рис. 2.1.а, де діапазон яскравостей має межі $x_{\min} = 180$, $x_{\max} = 240$.

При лінійному контрастуванні використовується лінійне поелементне перетворення виду: $y = a \cdot x + b$, параметри якого a і b визначаються бажаними значеннями мінімальної y_{\min} і максимальної y_{\max} яскравості. Вирішивши систему рівнянь:

$$\begin{cases} y_{\min} = a \cdot x_{\min} + b \\ y_{\max} = a \cdot x_{\max} + b \end{cases}$$

відносно параметрів перетворення a і b , неважко привести систему до виду:

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} (y_{\max} - y_{\min}) + y_{\min}$$

Результат лінійного контрастування вихідного зображення, представленого на рис. 3.1.а, наведено на рис. 3.1.б при $y_{\min} = 0$ і $y_{\max} = 255$.

Порівняння двох зображень свідчить про значно кращому візуальному якості обробленого зображення. Поліпшення пов'язане з використанням після контрастування повного динамічного діапазону екрану, що відсутня у вихідного зображення.



а)



б)

Рисунок 3.1 – Приклад лінійного контрастування

Також розглянемо соляризацію зображення. При даному виді обробки перетворення $y = f(x)$ має вигляд $y = k * x * (x_{\max} - x)$, де x_{\max} – максимальне значення вихідного сигналу, а k – константа, що дозволяє управляти динамічним діапазоном перетвореного зображення. Функція, що описує дане перетворення, є квадратичної параболою, її графік при $k = 1$ наведено на рис. 3.2. При $y_{\max} = x_{\max}$ динамічні діапазони зображень співпадають, що може бути досягнуто при $k = 4 / x_{\max}$. Інший шлях нормалізації динамічного діапазону може полягати в застосуванні одного з методів контрастування, наприклад, описаного вище лінійного контрастування.

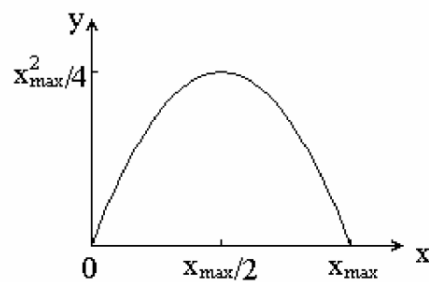


Рисунок 3.2 – Функція, що описує соляризацію

Як впливає з рис. 3.2, сенс соляризації полягає в тому, що ділянки вихідного зображення, що мають рівень білого або близький до нього рівень яскравості, після обробки мають рівень чорного.

При цьому зберігають рівень чорного і ділянки, що мають його на вихідному зображенні. Рівень ж білого на виході набувають ділянки, що мають на вході середній рівень яскравості (рівень сірого). Приклад застосування соляризації наведено на рис. 3.3. На рис. 3.3.а показано вихідне зображення, а на рис. 3.3.б – результат його соляризації. На другому етапі обробки тут застосовано лінійне контрастування при $y_{\min} = 0$ і $y_{\max} = 255$. Обробка призводить до підвищення чіткості деталей

зображення: поліпшені зображення очей, підвищений контраст на переході "особа - волосся" і т. д.



а)



б)

Рисунок 3.3 – Приклад соляризації

Наступним кроком є препарування зображення. Препарування являє собою цілий клас поелементних перетворень зображень. Характеристики застосовуваних на практиці процедур препарування наведені на рис. 3.4. Зупинимося на описі деяких з них. Перетворення з пороговою характеристикою (рис. 3.4.а) перетворює напівтонове зображення, що містить всі рівні яскравості, в бінарне, точки якого мають яскравості $y = 0$ або $y = y_{\max}$. Така операція, звана іноді бінаризацією або бінарним квантуванням, може бути корисною, коли для спостерігача важливі контури об'єктів, присутніх на зображенні, а деталі, що містяться всередині об'єктів або всередині фону, не уявляють інтересу. Основною проблемою при проведенні такої обробки є визначення порогу x_0 , порівняння з яким яскравості вихідного зображення дозволяє визначити значення вихідного зображення в кожній його точці. Найбільш виправданим для математичного опису зображення є застосування теорії ймовірностей, випадкових процесів і випадкових полів. При цьому визначення оптимального порогу бінарного квантування являє собою статистичну задачу. Статистичного підходу до обробці зображень в наступних розділах приділяється значна увага, в

тому числі і при вирішенні задачі розділення точок зображення на два класи – так званої бінарної сегментації.

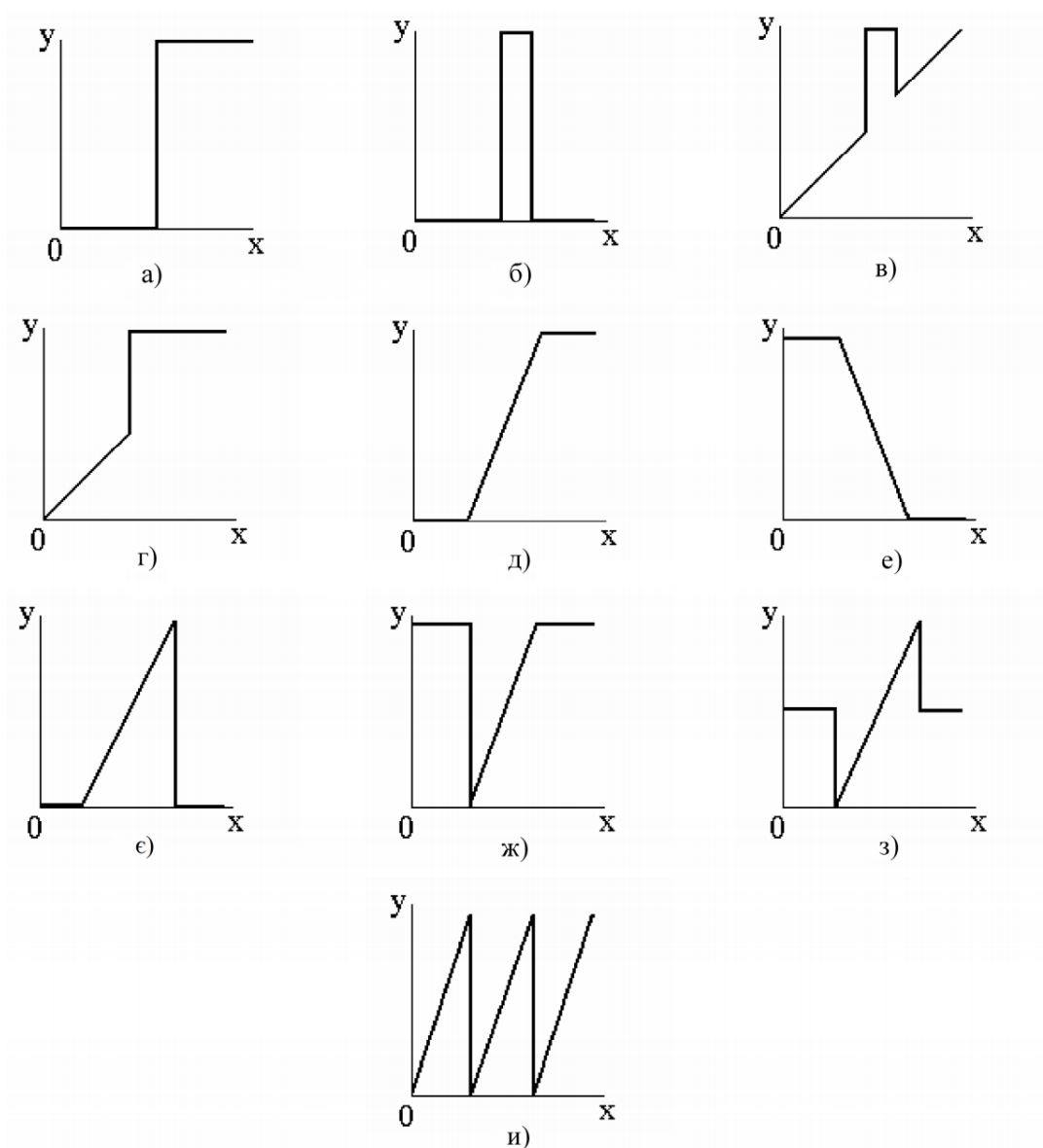


Рисунок 3.4 – Приклади перетворень, що використовуються при препаруванні

Тут же обмежимося обговоренням приватного, але практично важливого випадку. Іноді при обробці доводиться мати справу з зображеннями, збереженими як напівтонові, але за своїм змістом мало відрізняються від бінарних. До них відносяться текст, штрихові малюнки, креслення, зображення відбитка пальця, приклад якого наведено на рис. 3.6. а.

Щільність ймовірності $w(x)$, що описує розподіл яскравості такого зображення, може містити два добре розділені піки. Інтуїтивно зрозуміло, що поріг бінарного квантування слід вибирати посередині провалу між цими піками, як це показано на рис. 3.5.

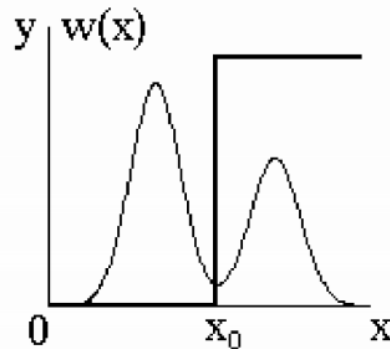


Рисунок 3.5 – До вибору порога бінарного квантування

Заміна вихідного напівтонового зображення бінарним препаратом вирішує два основні завдання. По-перше, досягається більша наочність при візуальному сприйнятті, ніж у вихідного зображення. По-друге, відчутно скорочується обсяг запам'ятовує для зберігання зображення, оскільки бінарний препарат для запису кожної точки бінарного зображення вимагає лише 1 біт пам'яті, в той час як напівтонове зображення для вирішення тієї ж задачі при найбільш часто вживаному форматі вистави - 8 біт. Приклад бінаризації зображення відбитка пальця наведено на рис. 3.6.б.

Сенс інших перетворень, представлених на рис. 3.4, неважко зрозуміти, розглядаючи їх характеристики. Наприклад, перетворення рис. 3.4.б виконує яркостний зріз зображення, виділяючи ті його ділянки, де яскравість відповідає виділеного інтервалу. При цьому інші ділянки виявляються повністю "погашеними" s (мають яскравість, відповідну рівню чорного). Переміщаючи виділений інтервал по яркостній шкалою і змінюючи його ширину, можна детально дослідити зміст картини.

Перетворення, наведене на рис. 3.4.є, також дозволяє підвищити детальність спостерігається картини в обраному діапазоні яскравостей, проте в відмінну від попереднього тут вихідне зображення використовує

повний динамічний діапазон. По суті, це перетворення являє собою лінійне контрастування, що застосовується до обраному діапазону вхідного зображення. Як і в попередньому варіанті, ділянки, що не потрапили в цей діапазон, утворюють після препарування чорний фон.

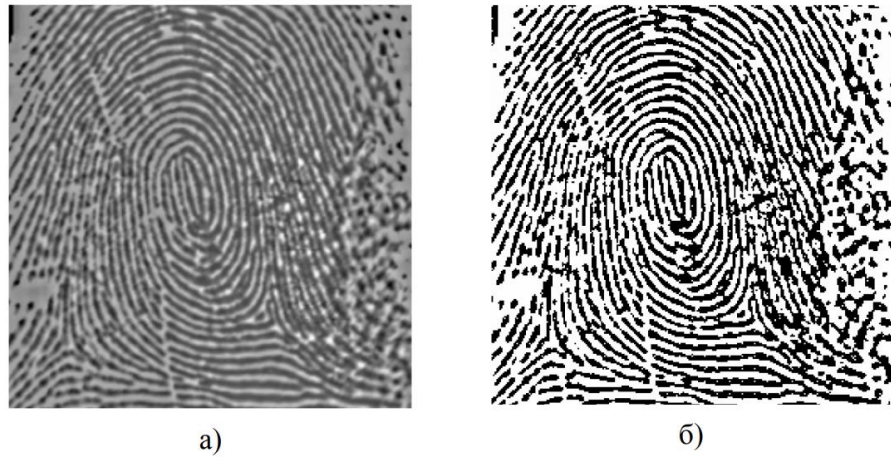


Рисунок 3.6 – Приклад бінаризації зображення

Іноді наочність зображення підвищується застосуванням перетворення типу пилоподібного контрастування s (рис. 3.4. и). При цьому різні яскравісні діапазони одночасно піддаються локальному яркостному контрастированию. Однак необхідно мати на увазі, що дане перетворення, як і деякі інші, може супроводжуватися появою помилкових контурів на одержуваному препараті.

На рис. 3.7 наведені результати експерименту, в якому до аерознімки ділянки землі s (рис. 3.7. а) застосовувалися перетворення типу порогова обробка (рис. 3.7.б) і пилоподібну контрастування (рис. 3. 7.в). Перше призводить до виявлення меж окремих ділянок, створюючи загальне інтегральне уявлення про спостережуваної сцені. Друге, навпаки, дає можливість спостереження дрібних деталей на всіх ділянках зображення. Поєднання двох таких можливостей може виявитися корисним спостерігачеві.

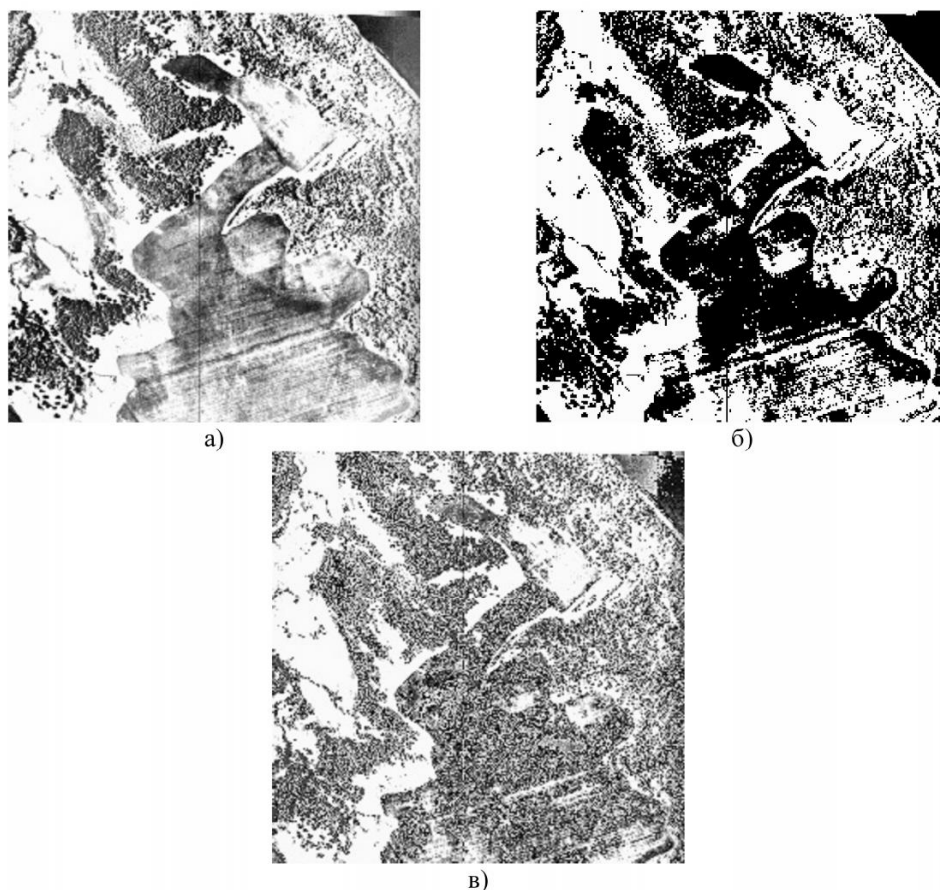


Рисунок 3.7 – Приклади препарування зображення

На закінчення відзначимо, що препарування часто використовується і в автоматичних системах обробки візуальної інформації, оскільки підготовлюваний при цьому препарат може містити всю інформацію, необхідну для подальшої (вторинної) обробки. Наприклад, якщо при спостереженні з космосу потрібно автоматично виявити на зображенні деякий об'єкт, який має відому конфігурацію, то для цього може бути досить бінарного препарату, що передає цю конфігурацію.

При всіх поелементних перетвореннях відбувається зміна закону розподілу ймовірностей, що описує зображення. Розглянемо механізм цієї зміни на прикладі довільного перетворення з монотонної характеристикою, описуваної функцією $y = f(x)$ (рис. 3.8), що має однозначну зворотну функцію $x = \Phi s(y)$. Припустимо, що випадкова величина підпорядковується щільності ймовірності $w_x(x)$. Нехай Δx –

довільний малий інтервал значень випадкової величини x , а Δy – відповідний йому інтервал перетвореної випадкової величини y . Попадання величини y інтервал Δy тягне за собою потрапляння величини x інтервал Δx , що означає вірогідну еквівалентність цих двох подій. Тому, враховуючи малість обох інтервалів, можна записати наближена рівність:

$$w_x(x)|\Delta x| \approx w_y(y)|\Delta y|$$

де модулі враховують залежність ймовірностей від абсолютних довжин інтервалів (і незалежність від знаків приростів Δx і Δy). Обчислюючи звідси щільність ймовірності перетвореної величини, підставляючи замість x його вираз через зворотну функцію і виконуючи граничний перехід при $\Delta x \rightarrow 0$ (і, отже, $\Delta y \rightarrow 0$), отримуємо:

$$w_y(y) = w_x(\varphi(y)) \cdot \left| \frac{d\varphi(y)}{dy} \right|$$

Цей вираз дозволяє обчислити щільність ймовірності продукту перетворення, яка, як видно з нього, не збігається з щільністю розподілу вихідної випадкової величини. Ясно, що істотний вплив на щільність $w_y(y)$ надає виконується перетворення, оскільки в формулу вище входить його зворотна функція і її похідна.

Співвідношення стають дещо складніше, якщо перетворення описується не взаємно-однозначною функцією. Прикладом такої більш складної характеристики з неоднозначною зворотною функцією може служити пилоподібна характеристика рис. 3. 4.и. Однак, загалом, сенс імовірнісних перетворень при цьому не змінюється.

Всі розглянуті в цьому розділі поелементні перетворення зображень можна розглянути з точки зору зміни щільності ймовірності. Очевидно, що ні при одному з них щільність ймовірності вихідного продукту не співпадатиме з щільністю ймовірності вихідного зображення (за винятком, звичайно, тривіального перетворення $y = f(x) \equiv x$. Неважко

переконалися, що при лінійному контрастуванні зберігається вид щільності ймовірності, проте в загальному випадку, тобто при довільних значеннях параметрів лінійного перетворення, змінюються параметри щільності ймовірності перетвореного зображення.

Визначення імовірнісних характеристик зображень, які пройшли нелінійну обробку, є прямим завданням аналізу. При вирішенні практичних задач обробки зображень може бути поставлена зворотна задача: за відомим увазі щільності ймовірності $w_x(x)$ і бажаного виду $w_y(y)$ визначити необхідне перетворення $y = f(x)$, якому слід піддати вихідне зображення. У практиці цифрової обробки зображень часто до корисного результату приводить перетворення зображення до рівноімовірними розподілу. У цьому випадку

$$w_y(y) = \begin{cases} \frac{1}{y_{max} - y_{min}}, & \text{при } y_{min} \leq y \leq y_{max} \\ 0, & \text{при інших } y \end{cases}$$

де y_{min} і y_{max} – мінімальне і максимальне значення яскравості перетвореного зображення. Визначимо характеристику перетворювача, вирішального це завдання. Враховуючи цей вираз, знаходимо:

$$F_y(y) = \int_{y_{min}}^y w_y(y) dy = \frac{y - y_{min}}{y_{max} - y_{min}}$$

Підставляючи цей вираз в умову ймовірнісної еквівалентності

$$F_x(x) = F_y(y)$$

після простих перетворень отримуємо співвідношення

$$y = (y_{max} - y_{min}) \cdot F_x(x) + y_{min}$$

представляє собою характеристику $y = f(x)$ в розв'язуваній задачі. Згідно з попереднім виразом вихідне зображення проходить нелінійне перетворення, характеристика якого $F_x(x)$ визначається інтегральним

законом розподілу самого вихідного зображення. Після цього результат приводиться до заданого динамічному діапазону за допомогою операції лінійного контрастування.

Аналогічним чином можуть бути отримані рішення інших подібних завдань, в яких потрібно привести закони розподілу зображення до заданого виду. Гіперболізація розподілу, передбачає приведення щільності ймовірності перетвореного зображення до гіперболічного виду

$$w_y(y) = \begin{cases} \frac{1}{y(\ln y_{max} - \ln y_{min})}, & \text{при } y_{max} \leq y \leq y_{min} \\ 0, & \text{при інших } y \end{cases}$$

Якщо врахувати, що при проходженні світла через око вхідна яскравість логарифмірується його сітківкою, то підсумкова щільність ймовірності виявляється рівномірною. Таким чином, відмінність від попереднього прикладу полягає в обліку фізіологічних властивостей зору. Можна показати, що зображення з щільністю ймовірності виходить на виході нелінійного елемента з характеристикою:

$$y = y_{min} \cdot \left(\frac{y_{max}}{y_{min}} \right)^{F_x(x)}$$

яка також визначається інтегральним законом розподілу вихідного зображення.

Таким чином, перетворення щільності ймовірності припускає знання інтегрального розподілу для початкового зображення. Як правило, достовірні відомості про нього відсутні. Використання для даних цілей аналітичних апроксимацій також малоприслужно, тому їх невеликі відхилення від істинних розподілів можуть призводити до істотного відмінності результатів від необхідних. Тому в практиці обробки зображень перетворення розподілів виконують у два етапи.

На першому етапі вимірюється гістограма вихідного зображення. Для цифрового зображення, шкала яскравостей якого, наприклад, належить цілочисловому діапазону 0...255, гістограма являє собою таблицю з 256 чисел. Кожне з них показує кількість точок у кадрі, що

мають дану яскравість. Розділивши всі числа цієї таблиці на загальний розмір вибірки, рівний числу використовуваних точок зображення, отримують оцінку розподілу ймовірностей яскравості зображення. Позначимо цю оцінку w_x^* , $0 \leq j \leq 255$. Тоді оцінка інтегрального розподілу виходить за формулою:

$$F_x^*(j) = \sum_{i=0}^j w_x^*(i)$$

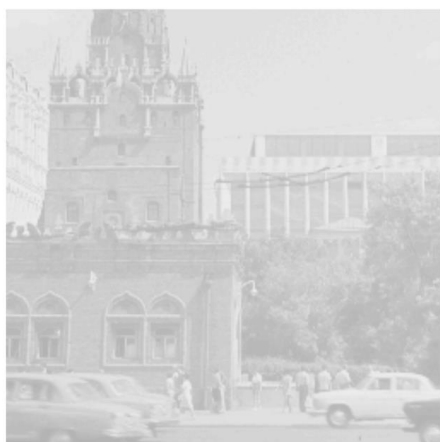
На другому етапі виконується саме нелінійне перетворення $y = f(x)$, забезпечує необхідні властивості вихідного зображення. При цьому замість невідомого істинного інтегрального розподілу використовується його оцінка, заснована на гістограмі. З урахуванням цього всі методи поелементного перетворення зображень, метою яких є видозміна законів розподілу, отримали назву гістограмного методів. Зокрема, перетворення, при якому вихідна зображення має рівномірний розподіл, називається еквалізацією (вирівнюванням) гістограм.

Відзначимо, що процедури перетворення гістограм можуть застосовуватися як до зображення в цілому, так і до окремих його фрагментах. Останнє може бути корисним при обробці нестационарних зображень, зміст яких істотно розрізняється за своїми характеристиками на різних ділянках. У цьому випадку кращого ефекту можна домогтися, застосовуючи гістограмну обробку до окремих ділянок.

Використання співвідношень вище справедливих для зображень з безперервним розподілом яскравості, є не цілком коректним для цифрових зображень. Необхідно мати на увазі, що в результаті обробки вдасться отримати ідеальний розподіл ймовірностей вихідного зображення, тому корисно проводити контроль його гістограми.

На рис. 3.9 наведено приклад еквалізації, виконаної відповідно до викладеної методикою. Характерною рисою багатьох зображень,

одержуваних в реальних зображують системах, є значна питома вага темних ділянок і порівняно мале число ділянок з високою яскравістю.



а) вихідне зображення



б) результат обробки

Рисунок 3.9 – Приклад еквалізації зображення

Еквалізація покликана відкоригувати картину, вирівнявши інтегральні площі ділянок з різними яскравостями. Порівняння вихідного (рис. 3.9.а) і обробленого (рис. 3.9.б) зображень показує, що відбувається при обробці перерозподіл яскравостей призводить до поліпшення візуального сприйняття.

3.2 Локалізація, Нормалізація, Сегментація

Системою автоматичного розпізнавання автомобільних номерів може бути програмний або апаратно-програмний комплекс, що виконує алгоритми автоматичного розпізнавання номерних знаків для автоматизації введення даних і їх подальшої обробки.

Автоматичне розпізнавання автомобільних номерів базується на наступних процедурах:

- Локалізація
- Нормалізація
- Сегментація
- Розпізнавання
- Синтаксичний аналіз

На рис. 3.7 наведена схема роботи системи розпізнавання.

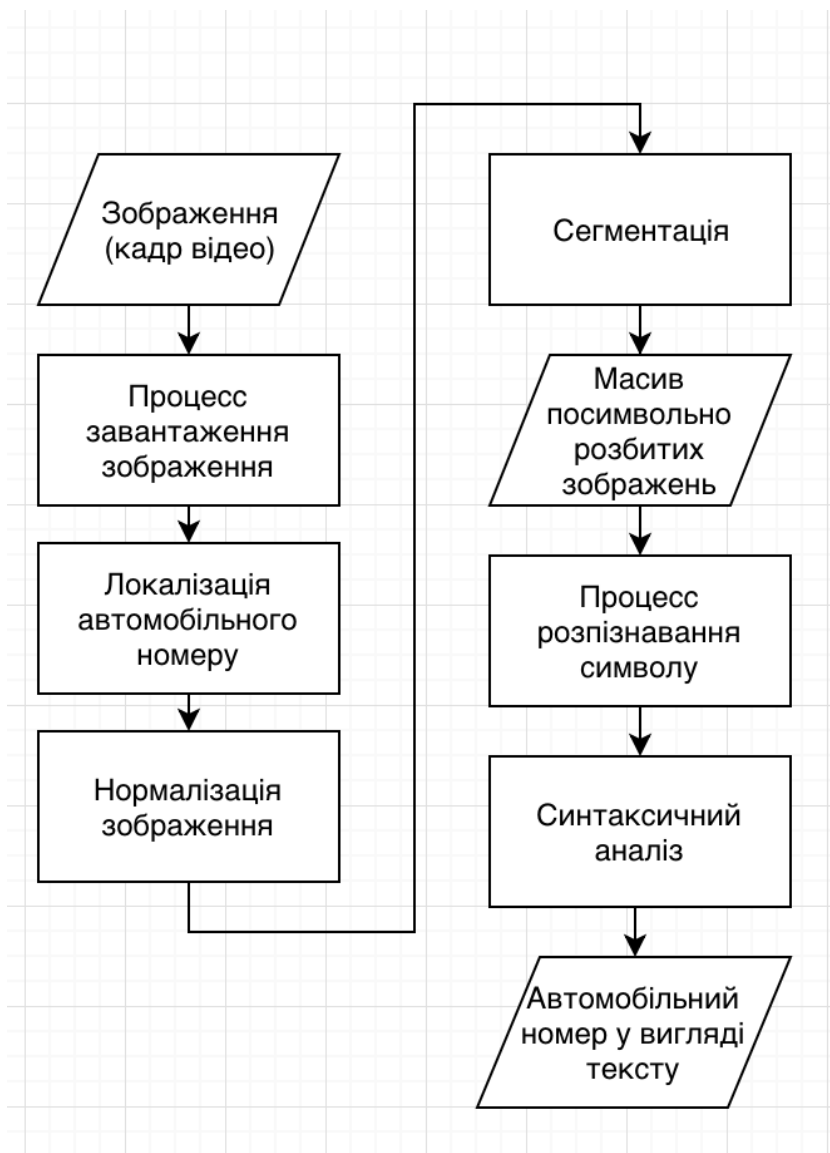


Рисунок 3.10 – Алгоритм роботи системи розпізнавання автомобільного номеру

Перша процедура призначена для виявлення і локалізації на зображенні області з реєстраційним номером автомобіля. Далі знайдена область вирізається з вихідного зображення і розглядається окремо. Нормалізація полягає в приведенні розмірів і орієнтації отриманого на попередньому кроці зображення з номером до необхідного для подальшої роботи виду, тобто до зображення номеру зі стандартизованими кутами повороту та нахилу тексту. Тут виконуються геометричні перетворення, шумозаглушення, зміна яскравості і контрасту. Процедура сегментації забезпечує поділ зображення на знакомісця, тобто виділення областей окремих символів. Процедура розпізнавання

призначена для формування рядка символів в автомобільного номеру. Остання процедура (процедура синтаксичного аналізу) виконується для визначення елементів рядка, що містить символи номера. Дані елементи можуть відрізнятися від відповідно до стандартів країни реєстрації автотранспортних засобів. Наприклад, в Україні з 2015 року використовується стандарт ДСТУ 4278:2012.

В якості вхідних даних будемо використовувати зображення знайдених номерних знаків згідно ДСТУ 4278:2012, а саме зображення номерних знаків для легкових і вантажних автомобілів, немаршрутних автобусів та мікроавтобусів, причепів, напівпричепів і причепів-розпусків. Розмір — 520×112 мм, тло — біле, символи — чорні. Поєднання літер ліворуч позначає регіон праворуч — серію номерного знаку (рис. 3.11).



Рисунок 3.11 – Номерний знак для легкових і вантажних автомобілів, немаршрутних автобусів та мікроавтобусів, причепів, напівпричепів і причепів-розпусків

Нормалізація зображення номерного знаку проводиться в два етапи. Спочатку визначається кут повороту номера в площині зображення. На другому – виконується алгоритм отримання нормалізованого зображення номера з вхідного зображення з урахуванням кута його повороту (рис. 3.12).

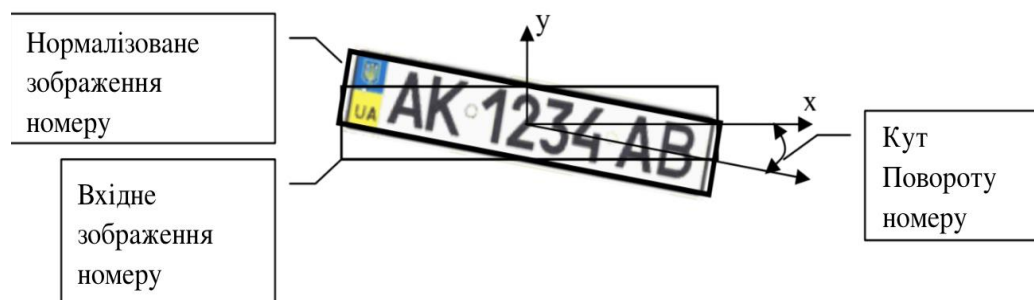


Рисунок 3.12 – Схема отримання нормалізованого зображення номера

Для повороту області зображення, що відповідає, нормалізованому номеру, використовується алгоритм, що базується на відповідному афінному перетворенні координат. Для зменшення потворень зображення при повороті, пов'язаних з його дискретним характером, використовується метод, заснований на білінійній інтерполяції по найближчих чотирьох пікселях.

Визначення кута повороту зображення номерного знаку виконується із використанням декількох етапів обробки та аналізу зображень. На першому етапі виконується операція підкреслення кордонів на зображенні на основі лінійного оператора Собеля для горизонтальних границь, має маску згортки:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

Наведений оператор більш чутливий до напрямів кордонів, близьким до горизонтального, тому дозволяє добре виділити на зображенні верхню і нижню частину номерного знаку, як показано на рис. 3.13.

На другому етапі виконується розрахунок карти щільності знайдених точок меж в просторі коефіцієнтів лінійних залежностей просторових координат згідно перетворенню Хафа. Метою другого етапу є визначення рівняння прямих, що відповідають верхній і нижній межі номерного знаку. Кожна точка карти кордонів, отриманої на попередньому етапі, породжує ціле сімейство прямих, що проходять через неї:

$$y_{ij} = ax_{ij} + b,$$

що в просторі лінійних коефіцієнтів також відповідає прямій:

$$b = x_{ij}a - y_{ij}.$$



а)



б)



в)

Рисунок 3.13 – а) фрагмент вихідного зображення зі знайденим положенням номера; б) вирізане зображення номерного знака з розширенням на 40%

у вертикальному напрямку; в) результат підкреслення кордонів. Наділяючи прями в просторі коефіцієнтів вагою $v(x_{ij}, y_{ij})$, відповідним значенням яскравості зображення результату підкреслення меж (Рис. 3в), і проводячи їх в просторі коефіцієнтів a і b з яскравістю, рівною вазі, отримаємо зображення, подібні до наведеного на рис. 3.14.

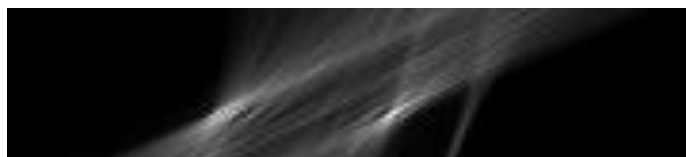


Рисунок 3.14 – Карта результатів перетворення Хафа

Вертикальне напрямок карти, наведеної на Рис. 4, відповідає змінам значень коефіцієнта a , горизонтальне - b . Діапазон зміни коефіцієнта розраховується виходячи з допустимих поворотів

зображення номера в площині зображення і розмірів знайденого зображення номера відповідно до формул:

$$a \in [-0,2; 0,2] \quad (0,2 \approx \operatorname{tg}(10^\circ)),$$

$$b \in [-a_{\max} W; H + a_{\max} W],$$

де a_{\max} - максимальне значення a , W - ширина зображення номера, H - висота зображення номера. Найбільш яскраві точки зображення карти результатів перетворення Хафа s (Рис. 31) відповідають найбільш яскравим прямим вихідного зображення кордонів (Рис. 30, в), що дозволяє визначити рівняння цих прямих і, відповідно, кути їх нахилу. Так як шуканими є прямі, відповідні верхній та нижній межах номерного знаку, то для визначення найбільш ймовірного кута повороту номера на зображенні карти результатів перетворення Хафа знаходяться дві точки, що мають одне значення a і різні значення b , з найбільшою сумарною яскравістю. Значення параметра a знайдених точок повністю визначає кут нахилу зображення номерного знаку ($a = \operatorname{tg}(\varphi)$).

Для сегментації символів на автомобільному номері пропонується використовувати підхід, заснований на підлаштуванні під реальне зображення різних моделей розташування символів на номері. Кожна з моделей відповідає определен- ному стандарту розташування символів. У даній статті розглядаються дві мо- поділи розташування символів в однорядкових номерних знаках, які можна перед- ставити наступними типами: {ББ цццц ББ} і {ЦЦЦ-ЦЦ ББ}, де Б - буква, Ц - цифра.

Кожну модель можна представити у вигляді зображення темних прямоуголь- ників, відповідних символів, на світлому тлі, як показано на рис. 3.15.

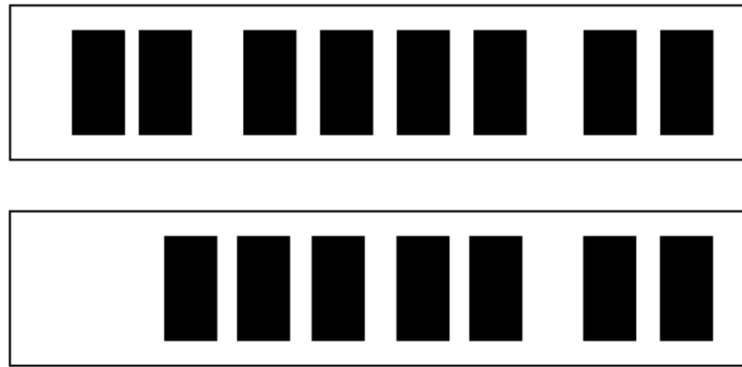


Рисунок 3.15 – Приклади використовуваних моделей розташування символів

Якщо номери мають інший тип, наприклад, білі символи на червоному тлі, або на чорному, то перед зіставленням моделі із зображенням останнім можна інвертувати по яскравості.

Як критерій відповідності моделі зображенню номера використовується величина:

$$K(x, y, W, H) = \frac{S_w - S_b}{\sigma_w}.$$

Тут σ_w - середня яскравість зображення під світлою областю, S_b - середня

яскравість зображення під чорної областю, σ_w - середній квадратичний розкид

яскравості зображення під світлою областю, x і y - координати моделі

всередині зображення номера, W і H - горизонтальний і вертикальний

розміри моделі. Чим більше значення критерію $K(x, y, W, H)$, тим

більше модель відповідає зображенню номера. Підгонка моделі під

зображення полягає у виборі згідно з прийнятим критерієм найкращого

положення і розмірів моделі. Після цього, згідно того ж критерію,

приймається рішення про найкращий тип моделі для теку- ного

зображення номера. Якщо враховувати можливі похибки при

нормалізації номера, на основі запропонованого критерію можна

незалежно визначити більш точне положення для кожного символу

окремо поблизу знайденого його положення з використанням всієї моделі.

Обчислення значення критерію $K(x, y, W, H)$ в кожному можливому положенні моделі всередині зображення номерного знака може бути ефективно виконано на основі використання інтегральних зображень для вихідного зображення номеру і зображення квадратів значень яскравості його пікселів.

Приклади спільної роботи послідовних етапів нормалізації зображення номера і сегментації символів наведено на рис. 3. 16. У верхньому рядку Рис. 3.16 показані вихідні зображення номерів, у другому рядку – результат їх нормалізації, в нижньому рядку – результат сегментації символів на номері (кожен символ являє собою окреме зображення).



Рисунок 3.16 – Приклади роботи алгоритмів нормалізації зображення номера і сегментації символів

Отримані після сегментації зображення символів дозволяють викорис застосовувати їх для вирішення задачі розпізнавання. Запропонований алгоритм сегментації дозволяє також визначити тип номерного знака, а отже, з'ясувати, є- ється чи кожен із символів буквою або цифрою, що дозволить полегшити аналіз зображень на наступному етапі розпізнавання. Запропоновані досить прості моделі дозволяють виконувати операцію сегментації з високою ефективністю за якістю і швидкості, а також дозволяють в разі потреби без значних додаткових

зусиль розширити число використовуваних моделей номерного знаку шляхом введення в розгляд інших можливих розташувань символів.

3. 2 Навчання та використання рекурентної нейронної мережі

Навчання нейронної мережі потребує великої кількості підготованих та розмічених даних. При навчанні нейронної мережі було використано пропріетарний набір даних з приблизно 150 мільйонами зображень. На жаль, зараз немає відкритих наборів даних такої розмірності, тому для навчання нейронної мережі у даній роботі було використано 500 тисяч зображень, які були отримані за допомогою змішування двох найбільших публічних наборів даних для досліджень. При навчанні нейронної мережі використовується алгоритм tripletlo. Фактично, ми прагнемо знайти таке відображення $f(x)$ зображення R^d на множину ознак, щоб квадратична відстань між всіма автомобільними номерами одного автомобіля, незалежно від стану та умов зображення, була мінімальна, тоді як відстань між парами зображень номерів різних автомобілів була велика.

Відображення представлене як $f(x) \in R^d$. оно відображає зображення x на d -вимірний Евклідовий простір. Додатково ми обмежуємо відображення так, щоб воно знаходилося на d -вимірній гіперсфері, тобто $\|f(x)\|_2 = 1$. Причини використання саме таких параметрів пояснені у роботі[32] у контексті метода “найближчого сусіда”. В даному випадку ми бажаємо упевнитись, що зображення x_i^a (якір) (anchor) деякої конкретної людини є ближче до усіх зображень x_i^p (позитивних) ніж до будь-якого зображення x_i^n (негативні) усіх інших людей. Візуально це показано на рис. 3.17.

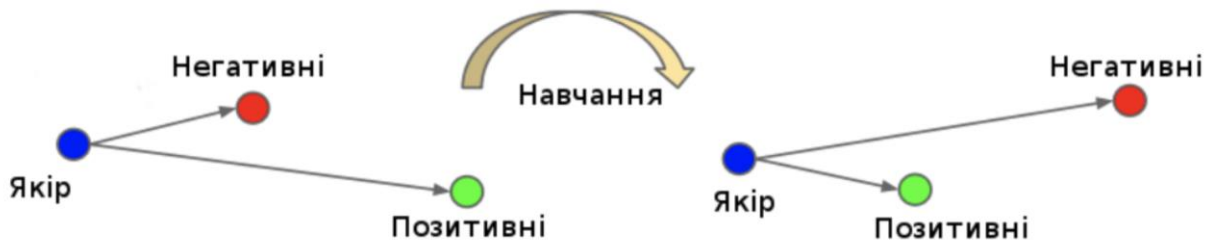


Рисунок 3.17 – Процес навчання за допомогою triplet lo

З математичної точки зору, нам потрібно, щоб була виконана нерівність

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2$$

$$\forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in T$$

Де α це мінімальна відстань між позитивними та негативними парами, T – це набір усіх можливих триплетів у тренувальному наборі даних, щонає потужність N . В такому випадку мінімізована втрата дорівнює:

$$L = \sum_i^N [\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha]$$

В результаті генерації усіх можливих триплетів буде багато триплетів, які одразу будуть задовольняти умові, що описана у формулі. Такі триплети не дають ніяких корисних результатів при застосуванні у процесі тренування і в результаті при пропуску крізь нейронну мережу будуть вести до більш низької збіжності. Дуже важливо обрати складні для нейронної мережі триплети s(тобто такі, які не задовольняють нерівність, які можуть сильно покращити модель. Це значить, що для даного ми хочемо обрати таке (дуже позитивне) s(hard poitive), щоб

$$\operatorname{argmax}_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2$$

Таким самим чином ми обираємо дуже негативне (hard-negative) значення таке, щоб

$$\operatorname{argmin}_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2$$

Неможливо порахувати argmin та argmax для усього тренувального набору даних. З цієї ситуації є два можливих виходи: Генерувати триплети офлайн кожні n кроків, використовуючи останні дані з мережі, та рахувати argmin та argmax лише на підвибірках даних. Генерувати триплети на льоту. Це може бути зроблено за допомогою вибору дуже позитивних/негативних прикладів з підвибірки. В даній роботі був використаний другий підхід з використанням великих підвбірок розмірністю декілька тисяч зображень та розрахунком argmin та argmax тільки для підвбірки.

Для того, щоб мати зрозуміле представлення відстані між якорем та позитивним прикладом, ми маємо бути впевнені, що деяка мінімальна кількість прикладів для кожної людини наявна у кожній підвбірці. В даній роботі використані такі тренувальні дані, щоб у кожній підвбірці було приблизно 40 зображень обличчя для кожної персони. Крім того, випадково обрані негативні приклади також додаються до кожної підвбірки.

Замість обирання дуже позитивних прикладів, використовуються усі пари якір — позитивний приклад у підвбірці, оскільки на практиці було виявлення, що використання такого підходу призводить до більшої стабільності тренування, та більшої збіжності на самому початку.

Як було сказано раніше, правильний вибір триплетів є критично важливим для швидкої збіжності. З однієї точки зору, краще було б використовувати малі підвбірки, адже завдяки цьому покращується збіжність під час використання методу стохастичного

градієнтного спуску. З іншої точки зору, особливості реалізації роблять використання підвбірок від десятків до сотень екземплярів більш ефективним. У більшості експериментів, що описані у розділі 4 були використані підвбірки розміром приблизно 1800 екземплярів.

4. ТЕСТУВАННЯ АЛГОРИТМУ РОЗПІЗНАВАННЯ ТЕКСТУ У СИСТЕМІ КОНТРОЛЬНО-ПРОПУСКНОГО ПУНКТУ

4.1 Особливості реалізації системи

Дана комп'ютерна система була реалізована у вигляді веб-додатку, що може бути застосований за допомогою браузеру (рис. 4.1).

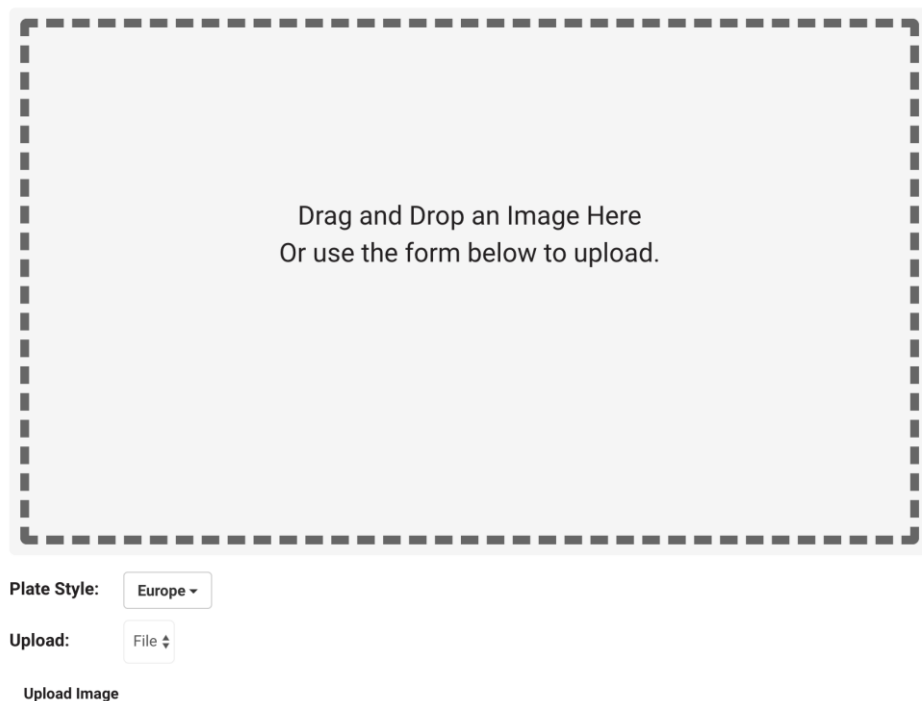


Рисунок 4.1 – інтерфейс веб додатку

Тренування системи відбувається наступним чином. Після запуску відповідної команди програма автоматично перевіряє у поточній директорії наявність директорії data/ що має містити тестові файли з тренувальним набором даних. Якщо вони є, то почнеться навчання системи. Після старту навчання програма починає виводити дані про поточний крок навчання з інформацію про похибку і затрачений час на певний крок, тощо s(рис. 4.2).

```
Run id: YGABLZ
Log directory: /tmp/tflearn_logs/

Training samples: 64
Validation samples: 64
--
Training Step: 207191 | total loss: 0.03896 | time: 2.469s
| Adam | epoch: 97191 | loss: 0.03896 - acc: 0.9905 | val_loss: 10.61265 - val_acc: 0.2500 -
- iter: 64/64
--
Training Step: 207192 | total loss: 0.04998 | time: 1.318s
| Adam | epoch: 97192 | loss: 0.04998 - acc: 0.9899 | val_loss: 10.59973 - val_acc: 0.2500 -
- iter: 64/64
--
Training Step: 207193 | total loss: 0.04510 | time: 1.319s
| Adam | epoch: 97193 | loss: 0.04510 - acc: 0.9909 | val_loss: 10.56708 - val_acc: 0.2500 -
- iter: 64/64
--
Training Step: 207194 | total loss: 0.04161 | time: 1.323s
| Adam | epoch: 97194 | loss: 0.04161 - acc: 0.9918 | val_loss: 10.55289 - val_acc: 0.2500 -
- iter: 64/64
--
Training Step: 207195 | total loss: 0.03909 | time: 1.315s
| Adam | epoch: 97195 | loss: 0.03909 - acc: 0.9911 | val_loss: 10.54458 - val_acc: 0.2500 -
- iter: 64/64
--
Training Step: 207196 | total loss: 0.04123 | time: 1.321s
| Adam | epoch: 97196 | loss: 0.04123 - acc: 0.9904 | val_loss: 10.54430 - val_acc: 0.2500 -
- iter: 64/64
--
Training Step: 207197 | total loss: 0.03826 | time: 1.321s
| Adam | epoch: 97197 | loss: 0.03826 - acc: 0.9914 | val_loss: 10.54394 - val_acc: 0.2500 -
- iter: 64/64
--
Training Step: 207198 | total loss: 0.03813 | time: 1.321s
| Adam | epoch: 97198 | loss: 0.03813 - acc: 0.9907 | val_loss: 10.54339 - val_acc: 0.2656 -
- iter: 64/64
--
Training Step: 207199 | total loss: 0.03444 | time: 1.318s
| Adam | epoch: 97199 | loss: 0.03444 - acc: 0.9916 | val_loss: 10.54625 - val_acc: 0.2656 -
- iter: 64/64
--
Training Step: 207200 | total loss: 0.03151 | time: 1.324s
| Adam | epoch: 97200 | loss: 0.03151 - acc: 0.9924 | val_loss: 10.52518 - val_acc: 0.2656 -
- iter: 64/64
--
Run id: NIY94X
Log directory: /tmp/tflearn_logs/

Training samples: 64
Validation samples: 64
--
Training Step: 207201 | total loss: 0.03787 | time: 2.492s
| Adam | epoch: 97201 | loss: 0.03787 - acc: 0.9916 | val_loss: 10.51123 - val_acc: 0.2500 -
- iter: 64/64
--
Training Step: 207202 | total loss: 0.03478 | time: 1.319s
| Adam | epoch: 97202 | loss: 0.03478 - acc: 0.9925 | val_loss: 10.48448 - val_acc: 0.2500 -
- iter: 64/64
--
[learn] 0:python* "ip-172-31-47-121" 14:22 02-Jun-17
```

Рисунок 4.2 – Вивід програми про поточний стан навчання нейромережевої системи

Усі дані про навчання автоматично записуються в теку /tmp/rflearn_log. Використовуючи ці дані, за допомогою CLI додатку Tensorboard можна проаналізувати, як працює система (рис. 4.3). Додаток обладнаний зручним та інтуїтивно зрозумілим інтерфейсом.

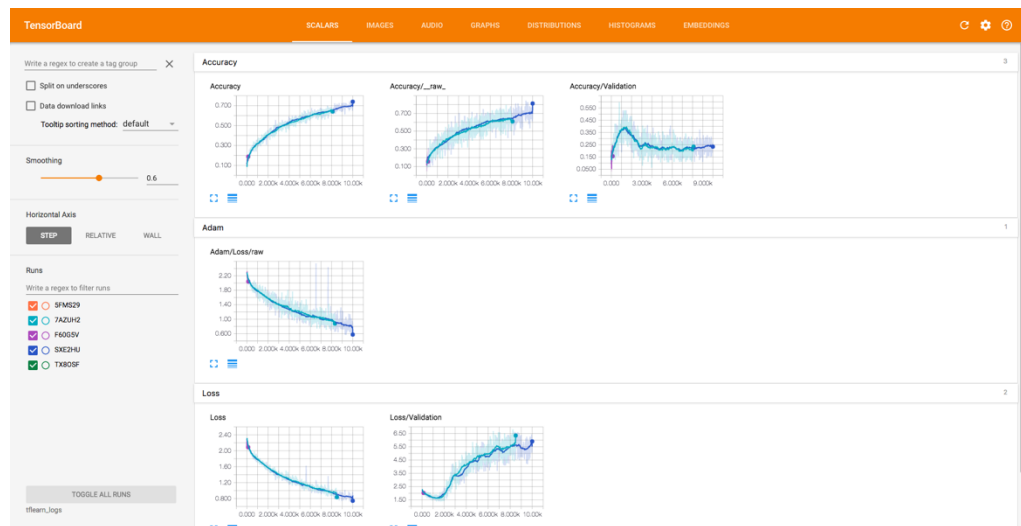


Рисунок 4.3 – Інформація про роботу нейромережевої системи.

Також можна детально роздивитися структуру нейромережевої системи, що була створена (рис. 4.4). Структура зображена у вигляді інтерактивного графу із згрупованими вершинами.

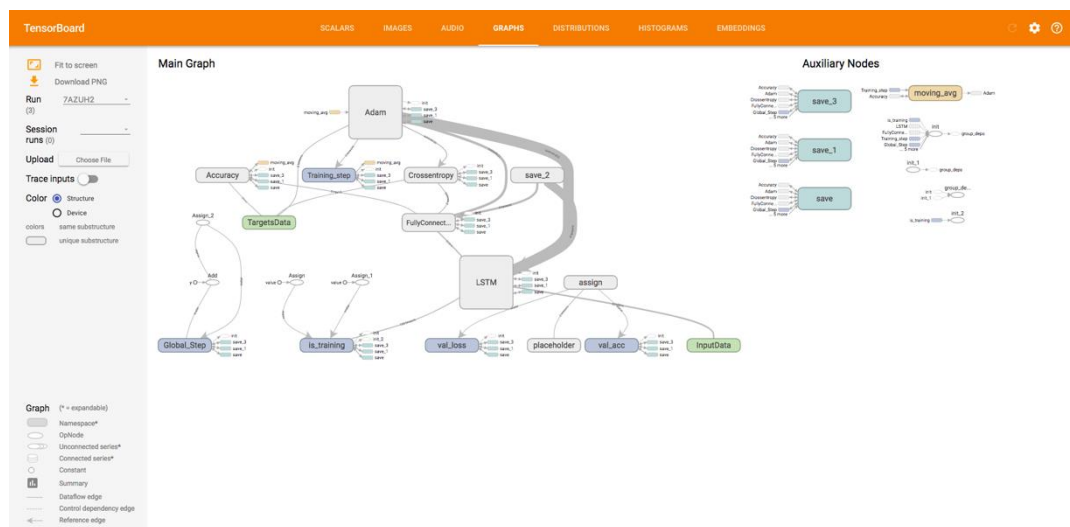


Рисунок 4.4 – Структура нейромережевої системи

Комп'ютерну систему розроблено на мові Python з використанням технології Tenorflow.

4.2 Тестування системи

В якості вхідного тестового набору візьмемо фотографії, що зроблені власноруч (рис. 4.5).



Рисунок 4.5 – Приклад фотографії з тестового набору

Після обробки фотографії системою отримуємо наступний результат зображений на Рис. 39. Якщо номерний знак, текст якого був успішно розпізнаний системою, є у базі, то користувач побачить *Access Allowed* – доступ дозволено. В протилежному випадку доступ надано не буде (рис. 4.6).



Рисунок 4.6 – Результат роботи системи



Рисунок 4.7 – Результат роботи системи

ВИСНОВКИ

В магістерській дисертації було розглянуто задачу розробки інформаційної системи розпізнавання автомобільних номерів в системі контрольно-пропускного пункту. В процесі вирішення отримано наступні результати:

1. Показано, що перспективним шляхом забезпечення достатньої якості розпізнавання автомобільних номерів являється вдосконалення математичного забезпечення інформаційної системи.

2. Розроблена архітектура інформаційної системи розпізнавання, що адаптована до застосування запропонованого математичного забезпечення..

3. Розроблене математичне та програмне забезпечення інформаційної системи.

4. Проведені експериментальні дослідження підтвердили перспективність застосування розробленого математичного забезпечення для розпізнавання автомобільних номерів.

5. Розроблену систему розпізнавання рекомендується впроваджувати в інформаційних системах загального призначення в яких є необхідність з помірною похибкою розпізнавати автомобільні номери.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. *Huruchi* Gurvey of Method for Character Recognition / G. Dedgaonkar uruchi, Anjali A. Chandavale, Ahok M. apkal // International Journal of Engineering and Innovative Technology (IJEIT). – 2012. - №5: [Електронний ресурс]. – Режим доступу: http://ijeit.com/vol%201/Issue%205/IJEIT1412201205_36.pdf
2. Знаки номерні транспортних засобів (Державний стандарт ДСТУ 4278:2004) [Чинний від 2004-02-20].–К.:Держстандарт України, 2004. – 22 с. – (Національний стандарт України).
3. *Paul Viola*. Robut real-time object detection / Paul Viola, Michael J. Jone // Proc. of IEEE Workhop on Statistical and Computational Theorie of Viion. – 2001.
4. *Гонсалес Р.* Цифрова обробка зображень /Р.Гонсалес, Р.Вудс – М. : Техносфера, 2006 – 1072с.
5. Fuhrman J.A., Yur'ev A.N., Yanhin V. Tifrovye metody obrabotki i rapoznavaniya binarnykh izobrazhenii [Digital method of proceeing and recognition of binary image]. Kranoyark, Kranoyark Univerity Publ., 1992, 248 p.
6. hin hi. Emgu CV Eential. Packt Publihing, 2013, 105 p.
7. Viola P., Jone M.J. Rapid Object Detection using a Booted Cascade of imple Feature. Proc. IEEE Conf. on Computer Viion and Pattern Recognition (CVPR 2001), 2001, vol. 1, p. 511–518.
8. Головкин Б.А. Машинное распознавание и линейное программирование.— М.: Советское радио. 1973. — 100 с.
9. Гренандер У. Лекции по теории образов. Т.1. Синтез образов. – М.: Мир, 1979. – 382с.
10. Патрик Э. Основы теории распознавания образов. — М.: Советское Радио, 1980. — 408 с.

11. Дэвид А. Форсайт, Джин Понс [Computer Vision: A Modern Approach Компьютерное зрение. Современный подход]. — М.: «Вильямс», 2004. — 928 с.
12. Джордж Стокман, Линда Шапиро [Computer Vision Компьютерное зрение]. — М.: Бином. Лаборатория знаний, 2006. — 752 с.
13. А. Л. Горелик, В. А. Скрипкин Методы распознавания М.: Высшая школа, 1989.
14. Ш.-К. Чэн Принципы проектирования систем визуальной информации М.: Мир, 1994.
15. В. Н. Вапник, А. Я. Червоненкис Теория распознавания образов М.: Наука, 1974. — 416 с.